GENERATING PARAPHRASED SENTENCES WITH VARIATIONAL AUTOENCODER MODELS

Engin Memiş, Elif Sena Yılmaz
Computer Engineering Department
Yıldız Technical University, 34220 Istanbul, Türkiye
{engin.memis, sena.yilmaz4}@yildiz.edu.tr

***Özetçe* —Veri, günümüzde büyük bir öneme sahip hale gelmiştir ve doğal dil işleme alanında kullanılan veri setlerinin çoğunluğu genellikle İngilizce dilinde oluşturulmaktadır. Türkçe dilindeki veri setleri bu veri setlerine göre daha az miktarda bulunmaktadır ve bu durum Türkçe dilinde gerçekleştirilen doğal dil işleme projeleri için bir engel teşkil etmektedir. Bu engelin kaldırılabilmesi için proje kapsamında Türkçe veri setlerindeki cümlelerden eş/benzer anlamlı cümleler üretilmesi amaçlanmıştır. Bu raporda, projenin amacı doğrultusunda çeşitli doğal dil işleme modelleri karşılaştırılıp uygun yöntem ile geliştirelen modelden üretilen cümleler kullanılarak Türkçe veri setlerinin zenginleştirilmesindeki başarısı incelenmiştir.**

***Anahtar Kelimeler*—doğal dil işleme, varyasyonel otokodlayıcı, cümle üretimi, transformatör, gizli uzay.**

***Abstract*—Data has become of great importance nowadays and the majority of data sets used in natural language processing are usually created in English. Turkish language datasets are less abundant compared to these datasets and this situation constitutes an obstacle for natural language processing projects in Turkish language. In order to overcome this obstacle, the project aims to generate sentences with same/similar meaning from sentences in Turkish data sets. In this report, various natural language processing models are compared for the purpose of the project and their success in enriching Turkish datasets is analyzed by using sentences generated from the model developed with the appropriate method.**

***Keywords*—natural language processing, variational autoencoder, sentence generation, transformer, latent space.**

I. Introduction

Language is the most important tool for communication. People use language to share ideas, feelings and thoughts with each other. However, understanding and processing language is not as easy a task for machines as it is for humans. Natural Language Processing (NLP), developed for this reason, is a technology consisting of a set of techniques and algorithms that enable human language to be understood and processed by machine.

NLP technology is also used to understand various grammatical structures in human language and make inferences. NLP technology is used in search engine query responses, social media analysis, text mining and many other areas.

A. Variational Autoencoders (VAE)

Variational autocoders (VAE) are one of the deep learning methods based on artificial neural networks and are widely used in Natural Language Processing (NLP). Unlike normal autocoders, it represents the sentence given as input by calculating a certain probabilistic distribution in the latent space. In this way, it can ensure that sentences are more diverse in the output.

VAEs, which are used to learn patterns and relationships in a given dataset, are very useful in tasks such as understanding grammatical structures in text data and generating alternative sentences. Therefore, the aim of this project is to generate synonymous sentences using VAEs.

II. Method

A. RNN-GRU Comparison

These comparisons were made using 100MB and 10MB datasets containing Turkish Wikipedia sentences.

Although GRU was predicted to run slower due to the gates in GRU, RNN was measured to run slightly slower. Regarding the quality of the output, it was observed that the quality of the sentences produced by RNN was very low compared to GRU. Therefore, RNN was not trained with the 100 MB dataset.

Table 1 Times Required for Training an Epoch (RNN-GRU)

	RNN	**GRU**
10 MB	102,80 sec	95,37 sec
100 MB	-	4470,34 sec

B. Word-Subword Comparison Using GRU

These comparisons were made using 100MB and 10MB datasets containing Turkish Wikipedia sentences.

The sentences in the dataset were processed with BERTurk's [1] 32K tokenizer to be used in the model. Since the total number of tokens in the dataset is reduced in the subword structure, a significant reduction in time is observed.

Table 2 Times Required for Training an Epoch (Word-Subword)

	Word	**Subword**
10 MB	93.37 sec	55.10 sec
100 MB	4470.34 sec	560.94 sec

C. Comparison of GRU-BERT Representations

BERT was trained one epoch from scratch with a 100 MB dataset. GRU was trained ten epochs from scratch with the same dataset. The difference in the number of epochs is due to the time it takes to train BERT. Representations from these trained models were compared on three different classification datasets. These are the dataset with three different news categories [2], the Twitter dataset with 5 different emotion categories [3], and the Twitter dataset with two categories as earthquake relief or not [4].

Table 3 Classification Success Rates of News Dataset Predicted by Representations

	SVM	Logistic Regression
BERT	%72,2	%73,7
GRU	%86,6	%84,6

Table 4 Classification Success Rates of Emotions Dataset Predicted by Representations

	SVM	Logistic Regression
BERT	%63,8	%65,5
GRU	%68,2	%65,7

Table 5 Classification Success Rates of Earthquake Relief Dataset Predicted by Representations

	SVM	Logistic Regression
BERT	%88,6	%90,6
GRU	%90,6	%91,3

D. Model's Architecture

As a result of previous comparisons, the model including the transformer structure was found to be more successful in sentence generation, and a transformer model including a variational autocoder was created.

1) Encoder: The model is designed in accordance with the transformer architecture and includes the following parts.

- Input Embedding
- Positional Encoding
- 6x Layer
    - Multi-Head Attention
    - Normalization Layers
    - Position-Wise Feed-Forward

2) Latent Space: After passing through the encoder, the given sentence is represented in a 16-dimensional latent space by adding noise. Sentences with similar meaning are expected to be located close to each other in this space.

3) Decoder: Again, the decoder is designed in accordance with the transformer architecture and includes the following parts. The point in the latent space represented by the sentence given as input to the encoder is taken and processed through layers in the decoder.

- Target Embedding
- Positional Encoding
- 6x Layer
    - Multi-Head Attention
    - Encoder-Decoder Attention
    - Normalization Layers
    - Position-Wise Feed-Forward

4) Loss Function: Below are listed some of the important parameters used in the training of this model.

- Batch Size: 64
- Learning Rate: 0.001
- Optimizer: Stochastic Gradient Descent (SGD)

E. GRU-BERT-Transformer Comparison

A ready-made model (BERT) containing the transformer structure to be used in the project was compared with other models. As a result of this comparison, the model containing the transformer structure was found to be more successful in sentence generation and a transformer model containing a variational autocoder was created.

The newly created model and other models were first trained on a 100 MB dataset. In order to test whether they are at the same level as the other models, their success was measured on various classification datasets.

Table 6 Success Rates of Models on Emotion Dataset

	SVM	Logistic Regression
GRU	%68,2	%65,7
BERT	%63,8	%65,5
Transformer	%68,3	%67,3

Table 7 Success Rates of Models on News Dataset

	SVM	Logistic Regression
GRU	%86,6	%84,6
BERT	%72,2	%73,7
Transformer	%68,1	%66,1

Table 8 Success Rates of Models on Earthquake Relief Dataset

	SVM	Logistic Regression
GRU	%90,6	%91,3
BERT	%88,6	%90,6
Transformer	%86	%88

F. Scalability of the Model

Since the success of the model reached the expected level, the scalability of the model was tested by increasing the training dataset by a factor of 10 to 1 GB and looking at the quality and training time of the sentences produced when trained for 1 epoch and the sentences produced when trained for 10 epochs with a 100 MB dataset. While doing this, the 1 GB dataset could not be trained in its entirety due to the insufficient RAM size. To solve this problem, the dataset was divided into 5 parts of 200 MB and the training result of each part was used as a checkpoint for training the next part. As a result of this comparison, training took about 8 hours in both cases, but the quality of the sentences was slightly better in the model trained with 1 GB. After these results, it was decided to continue training with the 1 GB dataset.

Input:	Türkiye'nin büyük bir otomotiv sanayisi vardır.
100 MB - 10 Epoch:	Türkiye'nin büyük bir a,si vardır.
1 GB - 1 Epoch:	Türkiye'nin büyük bir'-si vardır.

Input:	Mısır Tarihi boyunca 190 Kral hüküm sürmüştür.
100 MB - 10 Epoch:	ni vardır boyuncaen 1) 1.
1 GB - 1 Epoch:	Mısır a boyunca 190yın vetür.

Input:	Bu yolun ortalarına doğru, bozkıra açılan bir kapı vardır.
100 MB - 10 Epoch:	Bu : ortalarına doğru,in 2a'bir'vardır.
1 GB - 1 Epoch:	Bu de ortalarına doğru, diye aa 6 bir ne vardır.

G. Effect of Noise on Sentence Production

In order to ensure that the sentence generated by the model is not exactly the same sentence as the given sentence but a similar sentence, sentence generation is done by adding noise. However, since the model is not developed enough to produce the same sentences, it was deemed appropriate to remove the noise effect from the sentence generation.

Input:	Geç Eski Taş Çağı'nda Kuzey Afrika'nın kurak iklimi giderek ısındı ve kuraklaştı.
Without Noise:	Geç Eski birmı'nda Kuzey Afrika'nın ve iklimi nin olarakdı vee olan.
With Noise:	Avrupa Eski bir onunı'nda Kuzey Afrika'nın gibi iklimi nin olarakdı ve gibi).

Input:	Her şey diğer organizmalar ve çevreyle etkileşim içerisindedir.
Without Noise:	Her şey diğer dan 18 ve biryle ' içerisindedir.
With Noise:	Her şey diğer (18 ve çevreylema adı.

H. Effect of Epoch Number on Sentence Quality

Each epoch training took about 8 hours and sentences were generated from the model at the end of each epoch. When these sentences were analyzed, improvements in sentence quality were observed at the end of each epoch. Some examples of these sentences are shown below.

Input:	Takımı baskı altına almaya kimsenin hakkı yok.
1 Epoch:	dagn altına inl yok.
2 Epoch:	Kar baskı altına almaya - hakkı yok.
3 Epoch:	Takımı baskı altına almaya kimsenin hakkı yok.

Input:	Türkiye'de en çok aboneye sahip olan spor kulübü dergisidir.
1 Epoch:	Türkiye'de en çok :ye sahip olanleragndir.
2 Epoch:	Türkiye'de en çoklardanye sahip olan spor kulübügndir.
3 Epoch:	Türkiye'de en çok 3ye sahip olan spor kulübü dergisidir.

Input:	Hukuk biliminde biçim, öncelikler ve ilkeler doğrultusunda bazı sistemler ortaya çıkmıştır.
1 Epoch:	C büyük. (ği içinla 'tir bir tanım vardır).
2 Epoch:	Hukuk 2010de biçim, :r ve sonra doğrultusunda bazılara ortaya çıkmıştır.
3 Epoch:	Hukuk kısade biçim, öncelikler vele doğrultusunda bazıtu ortaya çıkmıştır.

Input:	Doktor Mortimer gözlük camlarının altında gözlerini kırpıştırarak şaşkınlıkla baktı.
1 Epoch:	ilenineer vardır :larının altındadıarakki olan W.
2 Epoch:	Aynı Mortimer vardır :larının altında.ıştırarak, olan baktı.
3 Epoch:	Doktor Mortimerg camlarının altında gözlerini sonraıştırarakkilıkla baktı.

III. Results

In this section, for performance measurement, we examined how the success rates change after the data set is enriched by generating the identical or similar sentences in the classification data sets with the model we have created and re-tested.

Sentence generation and enrichment for performance analysis were performed in three ways. These methods are as follows.

- **100% Production - 50% Add to Dataset:** For each sentence in the dataset, the corresponding similar sentence was generated and these sentences were ranked according to their success rate based on their loss rate. Then, half of the original dataset of these ordinary sentences were taken and added to the original dataset, i.e. the classification dataset was increased 1.5 times with the newly generated sentences.
- **100% Production - 100% Add to Dataset:** For each sentence in the dataset, the corresponding similar sentence was generated and then all sentences were added to the original dataset, i.e. the classification dataset was doubled with the newly generated sentences.
- **500% Production - 50% Add to Dataset:** For each sentence in the dataset, 5 similar sentences corresponding to this sentence were generated and then these sentences were ranked according to their success rate based on their loss rates. Then, half of the original dataset of these ordinary sentences were taken and added to the original dataset, i.e. the classification dataset was increased 1.5 times with the newly generated sentences.

Since the model has not been trained with a larger dataset over a longer period of time, it is not able to produce a successful response for each sentence due to the differences in sentence styles in the datasets from the sentence styles in the training dataset and the randomness that occurs during sentence generation. In the last mentioned method, it is planned to reduce the effects of this randomness by generating 5 sentences for each sentence and then taking only the most successful ones among them.

All sentence generation was done with the model we created after 1 GB 3 epochs of training. In addition to our own model, the BERT model [1], which was pre-trained with the Turkish 35 GB dataset available on the internet, was used for comparison while measuring the classification success. The tables below show the results for all three classification datasets after applying the above procedures.

Table 9 Success Rates Measured as a Result of Sentence Generation in Emotion Data Set

	Transformer		BERT	
	SVM	Log. Reg.	SVM	Log. Reg.
Original Dataset	%68,1	%63,6	%42	%59,2
100% Production - 50% Add to Dataset	%67,2	%61,8	%43	%56
100% Production - 100% Add to Dataset	%63,2	%57,1	%43	%51,7
500% Production - 50% Add to Dataset	%76,4	%65,6	%48,7	%59,5

Table 10 Success Rates Measured as a Result of Sentence Generation in News Data Set

	Transformer		BERT	
	SVM	Log. Reg.	SVM	Log. Reg.
Original Dataset	%67	%67	%82,3	%90,4
100% Production - 50% Add to Dataset	%70,2	%67,2	%83	%89
100% Production - 100% Add to Dataset	%70,4	%64,6	%78,4	%84,4
500% Production - 50% Add to Dataset	%76	%71,4	%81,3	%91,5

Table 11 Success Rates Measured as a Result of Sentence Generation in Earthquake Relief Data Set

	Transformer		BERT	
	SVM	Log. Reg.	SVM	Log. Reg.
Original Dataset	%88,6	%86,6	%92,6	%79,3
100% Production - 50% Add to Dataset	%89,2	%87,3	%88	%93
100% Production - 100% Add to Dataset	%89,7	%90,4	%75	%93
500% Production - 50% Add to Dataset	%95,4	%95	%93,3	%95,5

When the tables are analyzed, the decrease or increase in the first and second methods is due to the fact that the model is not always successful in generating sentences, as mentioned before. Since the number of sentences generated in the first method is less than the third method, and since all sentences generated in the second method are completely added back to the dataset, a decrease or sometimes a slight increase in the measurements in these rows is an expected change.

When we look at all the datasets enriched with the last method, it is an important result within the scope of this project that there is a noticeable increase in the classification success percentage compared to its original state.

IV. Discussion

Within the scope of this project, various models commonly used in natural language processing were compared and their results were tabulated. As a result of the comparisons, a transformer architecture model using a variational autocoder was created.

The model was tested and improved on various parameters. Afterwards, these data sets were enriched by generating synonyms/similar sentences from sentences in Turkish data sets. When the classification success was measured with the enriched data sets, an increase in the success rate was observed compared to the results of the original data set. These results support the goal of the project and can be further improved.

While training the model with a 1GB dataset, we had to train the model in parts due to the insufficient RAM size, only the first 60 tokens were taken because of the long processing time of overly long sentences, and it took about 8 hours to train even 1 epoch using the high-quality GPUs offered by Google for a fee.

Due to the limited resources available to us, not all of the sentences generated were of high quality, as the model could not be trained under the desired conditions. With a larger and more diverse dataset, the number of *epoch* can be increased to improve the quality of the sentences produced by the trained model. In this way, it is thought that the datasets can be enriched with a larger number of successful sentences.

In addition, a demo code has been written that allows the user to receive the output of the sentences generated with the same/similar meaning of the sentences given to the model in the form of a file.

References

[1] "Dbmdz/bert-base-turkish-cased · hugging face," Feb 2020. [Online]. Available: https://huggingface.co/dbmdz/bert-base-turkish-cased

[2] [Online]. Available: http://www.kemik.yildiz.edu.tr/veri_kumelerimiz.html

[3] A. Guven, "Turkish tweets dataset," 2021. [Online]. Available: https://www.kaggle.com/datasets/anil1055/turkish-tweet-dataset

[4] U. Kucuktas, "Turkey earthquake relief tweets dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/ulkutuncerkucuktas/turkey-earthquake-relief-tweets-dataset