Automatic Product Review Generation for Turkish

Emrah Doğan

2017400306

Supervisor: Tunga Güngör

Final report for CmpE
492 Senior Project

Department of Computer Engineering
Faculty of Engineering
Boğaziçi University

13 June 2022

Abstract

In today's world, especially during the pandemic period, online shopping has increased more than ever before. However, it may not be possible to examine some products on the internet, and people may expect more than necessary when they only look at the features of the product. However, when they encounter the product, they may be disappointed. This and similar situations can harm not only the users of the product but also the manufacturers. Because of this situation, users tend to buy products by looking at the comments of people who use the products, and it has become an important factor in the sales of the product. Because of this situation, it will be useful work not only for users but also for manufacturers to produce comments according to user types by looking at the features of the products.

In this project, I will try to produce a comment about the product that is compatible with them by looking at the number of stars given by a certain user for a certain product.

In this report, I will talk about my literature studies related to this project and my progress in the project.

Contents

Automatic product review generation for Turkish

Automatic product review generation for Turkish

Literature Review

1. Need for Product Review Generation

Recently, the serious increase in product sales has caused manufacturers to direct their attention to this area. Another change is that users can now easily make evaluations about products. As a result, people began to consider the user and expert comments instead of reading product descriptions, which could be misleading, to have an idea about the product. At the same time, companies started to examine the personal tastes of users as an important factor in whether they like the product or not, no matter how good or bad the product is. In addition, some of the comments focus on the features of the product. For this reason, in addition to being able to produce comments specific to user types, comments specific to product types have become very valuable not only for companies but also for users.

Companies will be able to use it to improve their products by predicting user reviews without producing the product. In addition, if users are undecided while choosing the products, even if there are very few people who use this product, they can make an easier decision by looking at the comments produced by this model.

2. Obstacles

There are many different problems that can be encountered in this study. First of all, we do not have a ready-made Turkish dataset because we want to do this study in Turkish.

Another challenge is that each product has different types of technical specifications. In addition, the number of comments describing the features of the products is very few. The quality of the comment content is more important than the number of comments. Moreover, some users may be malicious or prejudiced and make conscious or unconscious meaningless, bad or irrelevant comments about the products.

In addition, there may be differences in the choice of words according to the users. At the same time, the ratings given by the users' comments sometimes do not match.

3. The previous works

[1] worked on the generation of user and product-specific reviews that agree with input rating. This paper proposes a neural network that contains three parts: attribute encoder, sequence decoder, and an attention mechanism. Attribute encoder is a 1 hidden layer perceptron that encodes attributes from user id, product id, rating. Subsequently, the resulting encoded vector is fed into multi-layer LSTM which generates predicted review. Also, this paper introduces an attention mechanism that improves the performance of the paper.

In this paper, a data set was created on the books sold on Amazon between 1996-2014.

The deficiencies in this study are to create the product properties by taking only the ids of the products. Using only the id of the product does not give information about that product. In addition, although they claim that they obtained information about the product from the comments of the users, factors such as the character limitation during the dataset creation phase do not support their claims. The comments they produced are short comments that can be applied to almost all products and do not contain information about the product. Users, on the other hand, are most affected by the comments containing the details of the product.

Works Done

1. Creating a Dataset

Although we want to do this study in Turkish, datasets in Turkish are very limited. When I did dataset research in this area, I could not find a dataset suitable for our needs. As a result, I decided that I had to create the dataset myself.

First of all, I decided to start working on movies because movie reviews can be of much higher quality than other product reviews and it is possible to reach movie reviews. In other types of products, comments are generally very general and narrow, and product reviews are made over video.

First of all, I looked at the sites that fit my needs. The sites I consider suitable are as follows:

Movie review by editors:

https://cinepopularica.blogspot.com/
https://tarcinlikahve.com/category/sinema/film-incelemesi/
https://www.furkanozden.net/category/film-incelemeleri
https://filmloverss.com/kategori/elestiriler/
https://www.beyazperde.com/filmler/elestiriler-beyazperde/

Movie review by users:

https://www.beyazperde.com
https://turkcealtyazi.org

1.1 Dataset Creation Bot

I wrote a bot in Python for Beyazperde.com that gets the audience comments, audience names, and ratings in the best movies category.

I wrote 2 bots for Beyazperde.com, which received audience comments, audience names and ratings in the best movies category in Python. Despite the fact that the first version had a number of flaws, it was a good enough work to start. In the second version, I wrote this bot from scratch, taking advantage of my experience in this field in the meantime, and this version worked considerably more stable and faster, allowing me to download the entire site in a single day.

1.1.1 Version 1

Firstly, I used the BeuatifoulSoup library for HTML parsing. However, this site is written in Javascript Language. Therefore, the basic request function does not work because while parsing tag classes or attributes change. As a result of this, I used the Selenium library. Then I realized that there are a lot of inconsistencies in the tag classes, types, or attributes. I handled all inconsistencies. Afterward, I passed the measures, etc. taken by the site. For example, when I tried to go to the next page for the movie's comments, they were intentionally changing the content of the HTML page.

Sometimes they were showing the old version of the site. Sometimes due to links, sometimes due to intense requests. I solved intense requests problems. I didn't parse the old version of the site because it required almost as much programming to write a second bot.

There were also many minor difficulties. I do not want to continue the text by mentioning them here.

In summary, I created a dataset of 37836 comments from 219 movies in total. In addition, most of the comments here are detailed and quality comments with good information about the movie.

1.1.2 Version 2

In this version, without using Selenium, I just got the content of the pages with the requests and then parsed it using the Beautifulsoup and regex libraries.

This bot works much faster and more stable and also requires much less system requirements. The reason for this is that it does not perform browser-based rendering as in Selenium.

In this study, 232 336 Turkish movie reviews were obtained from 6972 movies. As stated in the paper, when we consider the products that have at least 10 comments for a product, this number drops to 215279.

1.2 English Amazon Book Reviews Dataset

The ready-for-training version of the dataset used in [1] was taken from its authors. In this dataset, there are 937 033 product reviews from 80 256 movies. The number of users is 19 675.

This data set is divided into 3 files as train, validation and test set, 70%, 10% and 20%, respectively.

2. Data Cleaning Pipeline

The paper [2] describes the stages of clearing various typos. In addition, you can try these processes on the samples you want on a website they have created. This model added to this website will be used to clean up the dirty dataset we have.

The planned steps to be used are as follows:

1. Tokenizer
2. Deasciifier
3. Vowelizer

When we look at the number of unknown words caused by misspellings in the Turkish data set we collected, data cleaning will make a positive contribution to the performance of the model. However, when we look at the performance of [2], it sometimes corrects the wrong word and sometimes turns the right word into the wrong word. Therefore, it is difficult to say that this cleaning work will contribute positively to the Turkish model without trying it.

2.1 Data Cleaning Bot

[2] is made available to people via the [3] site by the authors of the study, and they can also provide API Key specifically.

In addition to the steps given above, data cleaning was carried out with different steps using this study. However, neither result was satisfactory. Correct words were distorted and only some mistakes were corrected.

For this reason, studies were not continued with the data set that emerged as a result of this cleaning process.

2.2 Data Cleaning on Common Mistakes

Some of the errors in the comments in the dataset are very repetitive. By detecting these and mapping them as they should be, the same error was prevented from repeating in many samples.

Firsly the vocabulary is created then a dictionary is created with keys and values with same tokens(words). These tokens are checked manually, starting with the most common and fixed. After this checking process, the keys of this file were mapped to values in the dataset. By doing these, repetitive typos were corrected.

This new dataset is named as v2.2.

3. Making the dataset suitable for training

Reviews should be tokenized and special tokens should be added to the beginning and end of the token lists that emerge from here. In addition, special token (<unk>) creation should be added for unknown words. Moreover, for the comments to be batched, all comments to be batched must be padded to the same length.

Writing all this from the scratch is quite long and tedious. Therefore, all the necessary steps to prepare the data for training were done using the "torchtext" library.

First of all, the data must be read correctly. Then, the reviews are then tokenized using the Spacy library. The beginning of the sentence (<bos>), the end of the sentence (<eos>) tokens are added automatically while creating batches. For creating a vocab, there is a function in Torchtext. After creating a vocabulary, the embedding vectors of those words should be loaded to match correctly.

During the batching process, words that are not in the vocabulary will be replaced with the unknown token (<unk>).

In addition, while batching, batching samples with close comment lengths will be a much more efficient method. Since there will not be much difference in length between the longest sample in these batches and the other samples, the number of padding tokens that need to be added will be less. This will both reduce the load on the CPU and reduce the average sentence length that needs to be processed on the GPU. BucketIterator provided by Torchtext provides this to users.

In addition, a vocabulary consisting of user and product names were created and tokenized. Those that are less frequent than a specified number fall into the category of the unknown and this products are removed from the data.

3.1 English Amazon Book Dataset Preperation

There are three separate files separated by tabs in txt file format. Since this dataset is a dataset created by the authors of the article by clearing typos and emojis and selecting comments with less than 50 words, there was not as much work for us as in the Turkish dataset.

Moreover, words in comments are tokenized, separated by spaces. But the only mistake here is that sometimes words have punctuation marks combined at the end. In other words, words and punctuation marks are not separated by spaces. Such errors were corrected before starting the training.

3.2 Turkish Movie Dataset Preperation

While the necessary information was being taken from the site, a small amount of html codes were also inserted due to the inconsistencies on the site. This situation caused meaningless words to enter the vocabulary while creating vocabulary. First of all, these were detected and cleaned.

Moreover, there were many types of emoji in the comments, they were cleared. In addition, various punctuation marks were used more than once at the same time, for example “..." . Also often there were punctuation marks between words but with no space to separate them. All of these cases were detected and cleared using Regex.

After cleaning, the vocabulary was reconstructed and the words at vocabulary were looked at again. It was observed that the new words that emerged as a result of lowering the letters of the words containing various capital Turkish special characters in the comments could not be reduced correctly. For example, when the word "GELDİM" is converted to lowercase letters, it is translated as "geld¡m". The "¡" character is the problematic part here. As a result of such translation, two different words "geldim" and "geld¡m" are created in the vocabulary. This problem has also been fixed.

Another problem in this Turkish dataset, which is not in the English dataset, is the separation of various ownership suffixes. For example, the words “Hector’ a", “Hector’ un" are perceived as a single token. Such situations are mentioned so often in the comments that they can be entered into the vocabulary as a word. Moreover, a token can be classified as an unknown word (<nonk>) when it cannot enter the vocabulary (Word frequency less than 15.). However, if it is disassembled correctly, it will not be a problem. For example “Hector’un” can be transformed as [“Hector”, “‘“, “un”]. This problem was solved in the Parsing phase right after the reading the data and before giving the data to the model.

3.2.1 Dataset Version 2.0

At the stage of obtaining the comments from the site, some words were obtained together because they did not leave any spaces after the html tags were deleted. After this situation was determined, all the data were drawn correctly again and it was named as the 2nd version of the Dataset.

3.2.2 Dataset Version 2.1

In this version, reviews with a token length longer than 300 were not included in the dataset.

The reason for doing this is that the model has difficulty in long interpretations. Although the LSTM structure is good for long sequential data according to Vanilla RNN, this situation is not valid for a certain length.

3.2.3 Dataset Version 2.2

In this version, not only reviews with a token length longer than 300 were not included but also some common typos are manually corrected in the dataset.

Details are described in section Dataset Version 2.2.

The motivation for correcting some common spelling mistakes is when there are multiple spellings of the same word, the accompanying words and the meanings that can be derived from them are reduced. For this reason, it is thought that the meaning of the word cannot be learned well enough.

In addition, frequently used words have different spelling mistakes in too many repetitions. This also increases the number of words that need to be guessed. This will negatively affect the performance of the model.

4. Data Analysis

Since the ENG-Amazon data set in the reference document and the Turkish data set are collected from different languages and sources, there may be some differences between them. These differences can seriously affect the performance of the models.

Therefore, comparing the important features of the two data sets will provide us with important information about the problem.

Turkish Train, Valid, Test set MEAN review lenghts :

(48.96143941214847, 49.20663408380739, 47.81977326443136)

English Train, Valid, Test set MEAN review lenghts :

(35.98441001686145, 36.03853665304206, 36.35527439219621)

Turkish Train, Valid, Test set MEDIAN review lenghts:

(29.0, 29.0, 29.0)

English Train, Valid, Test set MEDIAN review lenghts :

(33.0, 33.0, 34.0)

In addition, when we look at the distribution of review lengths for Turkish and English datasets, we can easily say that the lengths of comments in the Turkish dataset are much more variable. However, this does not apply to the English dataset.

Also, while the maximum token length of comments in the English dataset is 81, this number increases to 4407 in the Turkish dataset. While it was stated in the reference article that they deliberately did not receive long comments because they contain detailed information about the product, we did not comply with them.

In addition, while the Batch size is 2 during the training with the Turkish dataset, this number is 128 in the English dataset for 16 GB GPU RAM. The reason for this difference is the very long comments in the Turkish dataset.

Examples above a certain number of comment tokens number in the Turkish Dataset may not be included in the training.

With minimum frequency 10:

Turkish Train, Valid, Test set MEAN (<unk> token sayısı)/(review length) ratio:

(0.09084741, 0.092510134, 0.09322953)

English Train, Valid, Test set MEAN (<unk> token sayısı)/(review length) ratio:

(0.0080943825, 0.008402237, 0.008261658)

There is a 10 times difference. In the Turkish dataset, one of every 10 tokens corresponds to the token. This is a relatively high rate. This high token rate can be cited as a reason for entering the token generation cycle.

The most logical comment to come out of the Turkish model so far was "çok sıkıcıydı ."

5. Implementing the Model

The model was written from scratch without using pre-written code. The structures in the earlier version of the code were not functional, or modular in other words.

In the new version, the code is divided into three parts: Encoder, Decoder and Attention. The advantage of dividing it into three parts in this way is that the variables of that function are deleted from GPU RAM since the function is exited after the operations with the relevant part are finished. With the more effective use of GPU RAM, we were able to increase the Batch Size, allowing us to see the training results more quickly.

Moreover, since Teacher Forcing was applied during the training phase in the previous version of the code, the operations can be performed in parallel. However, this situation also brought a serious disadvantage. Since the predicted word at the time of Inference was dependent on the previous words (autoregressiveness), operations could not be performed in parallel. This caused separate code writing for training and inference. The workload doubled when there was a change in the model. In the new version of the code, one word is predicted for each time step. Although the parallelism during training is lost, the need to write separate code for training and inference is eliminated.

In addition, for each word in the generated comment, a visualization was added showing how much the attention mechanism used in the model chose, giving weight to the user ID, movie ID and rating entries, respectively.

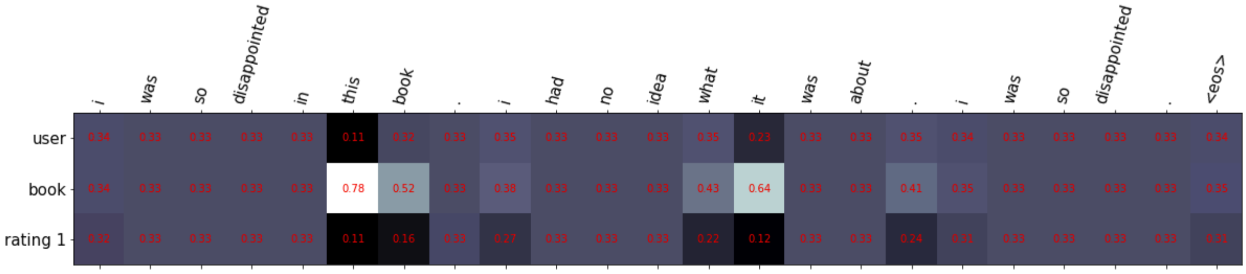

Fig 1. Attention scores for each token in generated review sample

6. Performance Metrics

Since BLEU Score was used as the main performance metric in the reference article, this metric was also preferred in this study.

6.1 BLEU Score

The Bilingual Evaluation Understudy Score, or BLEU for short, is a metric for evaluating a generated sentence to a reference sentence. Perfect match score is 1.0 whereas perfect mismatch score is 0. This metric was originally designed for Machine Translation tasks.

Advantages of BLUE Score Metric:

- Fast and inexpensive to calculate
- Language independent
- Easy to Understand
- Correlate well with human judgment on many generation tasks

The BLEU score measures the precision of n-gram matching by comparing the generated results with references, and penalizes small lengths by using a brevity penalty term.

6.1.1 Disadvantages of BLUE Scores

Considering the suitability of this metric for the task at hand, it is a measurement tool worth using, although it has some disadvantages.

The first drawback is that it needs a small vocabulary as it is sensitive to word choice. As the number of words in the vocabulary increases, the chance of matching the n-grams in the texts will decrease and unnecessarily bad results will appear.

Another disadvantage is the unknown words in the reference text. Having an unknown word in the reference text means that every word that can correspond to it will be accepted as correct. While the words corresponding to unknown tokens matching in the same position in the reference and generated text may be completely opposite to each other, the score will be high instead of low because BLEU score cannot distinguish this. This will mislead anyone looking at this metric.

Finally, it is very likely to express an idea that is desired to be expressed in the same idea by using very different word combinations. However, this metric will not be able to tell the difference as it is based on phrase word matches.

6.1.2 Implementation of BLEU Score

Since the BLEU Score is a very common metric used in NLP, this metric has been implemented in many libraries such as Torchetext and NLTK.

There should be absolutely no spaces in the tokens given when using the function in Torchtext. Otherwise, the function gives an error and it can be very difficult to solve because no informative message is returned.

When BLEU scores are calculated for all samples in the test data set one by one, the process takes too long to use. For this reason, making predictions as Batch instead of making predictions for each sample significantly shortens the time.

The process that requires effort to implement here is that each comment in the batch is brought to the same length by adding <pad> tokens, so it would be wrong to give them to the BLUE score as they are. First of all, the pad tokens here must be cleared. Then, Tensors of Int type should be replaced with their word equivalents and given to the BLEU score function.

In this study, these operations can be performed effectively and results can be obtained for high-dimensional datasets in a short time. In addition, as the batch size increases, the total time required to see the result decreases, which is another proof that this implementation accelerates.

7. Experiment Results

Experiment config 1 :

Min_word_frequency= 10, Max_Vocab_size= 15000, Hidden_size = 512

Dataset = v2.0

Experiment config 2 :

Min_word_frequency= 10, Max_Vocab_size= Unlimited, Hidden_size = 512

Dataset = v2.0

Experiment config 3 :

Min_word_frequency= 7, Max_Vocab_size= Unlimited, Hidden_size = 512

Dataset = v2.0

Experiment config 4 :

Min_word_frequency= 10, Max_Vocab_size= Unlimited, Hidden_size = 1024

Dataset = v2.0

Experiment config 5 :

Min_word_frequency = 10, Max_Vocab_size = Unlimited, Hidden_size = 512

Dataset = v2.1, Max_review_len= 300

Experiment config 6 :

Min_word_frequency = 10, Max_Vocab_size = Unlimited, Hidden_size = 512

Dataset = v2.2, Max_review_len= 300

Experiment config 7 : (Residual Connections)

Min_word_frequency= 10, Max_Vocab_size= Unlimited, Hidden_size = 512

Dataset = v2.1, Max_review_len= 300

Experiment config 8: (Residual Connections)

Min_word_frequency= 10, Max_Vocab_size= Unlimited, Hidden_size = 512

Dataset = v2.2,   Max_review_len= 300

7.1 Model for English

Best score is obtained at Epoch 7. There is a small difference between the BLEU score results stated in the paper and what we obtained. We think that this difference is due to minor differences in measurement methods. While we implement everything from scratch in the Lua language used by the authors of the paper, we preferred to benefit from the libraries as much as possible in this study.

When we read the comments from the model, we saw that there were very good comments like in reference paper.

Our Study	Reference Paper
BLEU-1 : 26.42 %	**BLEU-1 :** 30.48 %
BLEU-4 : 4.41 %	**BLEU-4 :** 5.03 %

Example Review :   User ID: 20,   Movie ID: 7,   Rating: 1

“i was so disappointed in this book . i had no idea what it was about . i was so disappointed .”

7.2 Experiment config 1

Training is stopped early due to bad results and very long training time.

Epoch	BLEU-1	BLEU-4
8	(0.09708851924072597,	0.005382212578492811)
7	(0.10072117029683747,	0.008312022039558931)
6	(0.06103806654515183,	0.005377403862154407)
5	(0.13653081908812972,	0.010864252451701525)
4	(0.1190287436381105,	0.014683190189721853)
3	(0.11559363646228729,	0.01141102974701928)
2	(0.10710471374420509,	0.011108278765175904)
1	(0.05745660420363781,	0.006755754351078308)

BEST EPOCHS :

Epoch 4   Epoch 5

BLEU-1 : 11.90 %	**BLEU-1 :** 13.65 %
BLEU-4 : 1.46 %	**BLEU-4 :** 1.08 %

Example <unk> token generation cycle :

rmel gibson ' ın <unk> <unk> <unk> <unk> …… <unk>

7.3 Experiment config 2

Training is stopped early due to bad results and very long training time.

Epoch	BLEU-1	BLEU-4
8	(0.106146403505199,	0.01149036764561244)
7	(0.11764161867404194,	0.009562498425143109)
6	(0.09443470285028199,	0.010032402786565685)
5	(0.1236291277007911,	0.011104855887427337)
4	(0.10795939143437529,	0.010906090863244766)
3	(0.10224247323935919,	0.009368193749618383)
2	(0.08232576573760435,	0.005266589730438863)

BEST EPOCHS :

Epoch 5	Epoch 7
BLEU-1 : 12.36 %	**BLEU-1 :** 11.76 %
BLEU-4 : 1.11 %	**BLEU-4 :** 0.95 %

7.4 Experiment config 3

Training is stopped early due to bad results and very long training time.

Epoch	BLEU-1	BLEU-4
8	(0.10693145278655376,	0.01150447619235511)
7	(0.10069089692774884,	0.010461897474878112)
6	(0.09451218195461522,	0.010838511880210452)
5	(0.1210199802545751,	0.011469460919749609)
4	(0.10076042410458369,	0.011599618553885218)
3	(0.09525158593590159,	0.010792355453066706)

BEST EPOCHS :

Epoch 5

BLEU-1 : 12.10 % **BLEU-4 :** 1.14 %

7.4 Experiment config 4

Training is stopped early due to bad results and very long training time.

Epoch	BLEU-1	BLEU-4
6	(0.07035928437389408,	0.003710317752400346)
5	(0.10708391161589192,	0.008458543487179893)
4	(0.1020429881447922,	0.007143422896598306)
3	(0.09303208178702378,	0.007617522342363771)
2	(0.0795379432798541,	0.005174211095727805)

BEST EPOCHS :

Epoch 5	Epoch 7
BLEU-1 : 12.36 %	**BLEU-1 :** 11.76 %
BLEU-4 : 1.11 %	**BLEU-4 :** 0.95 %

7.5 Experiment config 5

Epoch	BLEU-1	BLEU-4
1	(0.04247689803245588,	0.0030046239236447975)
2	(0.10310020546707944,	0.011953470474932726)
3	(0.11276849462698657,	0.013730175018878193)
4	(0.10080605134253368,	0.01239210024592982)
5	(0.11856712815069873,	0.015366771296350184)
6	(0.12606018206369007,	0.015373712828915658)
7	(0.10812787592860898,	0.014093409983814417)
8	(0.08940628287044979,	0.012770377226969627)
9	(0.12928549190903343,	0.015754177752993606)
10	(0.10876796377367888,	0.01453257602955588)
11	(0.09849235885354106,	0.012705932063876544)
12	(0.100588674746878,	0.014345367295689818)
13	(0.05890819639746015,	0.008758263346381776)
14	(0.146511014936012,	0.01674644474807097)
15	(0.12359678780665101,	0.014802047650300779)
16	(0.1149197515700811,	0.014771050337916402)
17	(0.12977345665269746,	0.016306270315097883)
18	(0.14263107660268454,	0.01776414697781971)

19	(0.13665925493225609,	0.017340104396947765)
20	(0.10860736077309703,	0.012932011755044187)

BEST EPOCHS :

Epoch 14

BLEU-1 : 14.65 %

BLEU-4 : 1.67 %

Epoch 18

BLEU-1 : 14.26 %

BLEU-4 : 1.77 %

7.6 Experiment config 6

Epoch	BLEU-1	BLEU-4
1	(0.10443794300796681,	0.009673708031597434)
2	(0.0976218080940092,	0.011415819898178648)
3	(0.12878912964057235,	0.01441649525149063)
4	(0.08288932084940985,	0.010870268864546591)
5	(0.10691385664707541,	0.013680706817200055)
6	(0.13239135344090666,	0.015007606865168038)
7	(0.12032389353974064,	0.014223412506859176)
8	(0.12051115559487825,	0.014601284218036093)
9	(0.14250361789179067,	0.017589835048445607)
10	(0.13658733581167434,	0.016124713559117293)
11	(0.12194241223491883,	0.014730475928340795)
12	(0.13778952372294864,	0.016251583146686177)
13	(0.10132745531368544,	0.013741285604660905)
14	(0.14025245468760078,	0.016863023104507213)
15	(0.13652786653577173,	0.016853424193837018)
16	(0.1279486719319924,	0.016164135236245433)
17	(0.13666530402966293,	0.017408674848396772)
18	(0.14065948030428732,	0.0171524239586606)
19	(0.1365581909036117,	0.01743941777710407)
20	(0.11328177703414337,	0.014878699756376535)

BEST EPOCHS :

Epoch 9

BLEU-1 : 14.25 %

BLEU-4 : 1.75 %

7.7 Experiment config 7

Epoch	BLEU-1	BLEU-4
1	(0.10439409351082427,	0.008154688161772257)
2	(0.10670826054203233,	0.010163330411171472)
3	(0.13703131241292812,	0.01389514556725908)
4	(0.09621000710675766,	0.011639315696444155)
5	(0.11852478324659144,	0.012612692921645912)
6	(0.1411498634206626,	0.014493724707949426)
7	(0.13231686312958416,	0.012988324532793857)
8	(0.13524460950118675,	0.01534855964323078)
9	(0.13326268538775166,	0.015175474120477093)
10	(0.11837679477917519,	0.012489790339267027)
11	(0.13288661493464718,	0.015070068166343669)
12	(0.13850085285040775,	0.015508825585778681)
13	(0.14689681025222429,	0.01648409227148831)
14	(0.14855309215295015,	0.015442941166722728)
15	(0.13537657052132668,	0.015030602822464106)
16	(0.14491884558442034,	0.016596960089776906)
17	(0.16095984068491878,	0.017211529967463314)
18	(0.14818131607898902,	0.016898324153464574)
19	(0.15146824854413798,	0.017230181684518966)
20	(0.1431405180842597,	0.015929427158883052)

BEST EPOCH :

Epoch 17

BLEU-1 : 16.09 % **BLEU-4 :** 1.72 %

7.8 Experiment config 8

Epoch	BLEU-1	BLEU-4
1	(0.09052654161508715,	0.012099110604250311)
2	(0.15066732847655429,	0.015775255890923663)
3	(0.10323984494682044,	0.01264445109066481)
4	(0.12756912127977718,	0.016888048907722164)
5	(0.039102032881778104,	0.005214072630989443)
6	(0.08585289632703737,	0.01131934195653223)
7	(0.08904445400688801,	0.011271946679412562)
8	(0.14282096837105807,	0.017029751411536613)
9	(0.1266760977603352,	0.0165048822346291)
10	(0.13839405933838725,	0.015581365981983496)
11	(0.14366985403541851,	0.016045514339532805)
12	(0.13148651706345924,	0.015568298227401339)

13	(0.14099991306584228,	0.015363322845697863)
14	(0.16021677246463234,	0.01724998709605285)
15	(0.1405859570308883,	0.016470075198377527)
16	(0.14530326397325458,	0.017025217601504487)
17	(0.16628621528384205,	0.01885324976720488)
18	(0.15755916921054608,	0.01884935880155308)
19	(0.15263370368069626,	0.01713287833852462)
20	(0.14072854677506383,	0.01644823222425077)

BEST EPOCH :

Epoch 16

BLEU-1 : 16.62 % **BLEU-4 :** 1.88 %

8. Example Generated Reviews

5 sample reviews from the top 3 models are as follows.

These comments were created by choosing the word with the highest probability at that time, as stated in the reference paper.

All extracted comments are in the supplementary file.

8.1 Experiment config 6

12, 12, 5-1
gerçekten çok iyi bir film . baştan sona kadar kendini izletiyor . baştan sona kadar insanı duygulandırıyor . <eos>

12, 12, 4-1
film gerçekten çok kaliteli . oyunculuk , senaryo , oyunculuk , senaryo hepsi mükemmel . 10 / 7 . 5 <eos>

12, 12, 3-1
film çok ağır ilerliyor ve insanı <unk> . ama yinede kendini izletiyor . <eos>

23, 4, 3-1
bu film hakkında çok fazla şey <unk> gerek yok . bana göre edward norton ’ un oyunculuğu için bile gidilir . <unk> <unk> ’ ın oyunculuğu çok yapmacık geldi bana . <eos>

23, 4, 2-1
bu film hakkında çok fazla şey <unk> gerek yok . <eos>

8.2 Experiment config 7

12, 12, 5-1
çok iyi bir film . ben 9 verdim . <eos>

12, 12, 3-1
çok iyi bir film . ben 6 verdim . <eos>

12, 12, 1-1
çok kötü bir filmdi . hiç <unk> . ben filmin sonunu <unk> hiç beğenmedim . <eos>

23, 4, 1-1
bu film hakkında söylenecek çok şey yok . hayatımda izlediğim en kötü filmdi . <eos>

23, 4, 5-1
bir çok kişi tarafından <unk> <unk> ama ben bu filmi çok beğendim ve çok güzel bir <unk> olduğunu düşünüyorum . brad pitt ' i hiç sevmem ama bu filmde çok iyi bir performans göstermiş . özellikle finali çok etkileyici ve şaşırtıcı . . <eos>

8.3 Experiment config 8

12, 12, 5-1
filmin sonu çok etkileyici . <unk> bir film . oyunculuk çok çok iyi . <eos>

12, 12, 3-1
filmin konusu , kurgusu , müzikleri , müzikleri , müzikleri , müzikleri , oyuncu performansları , müzikleri , karakterleri , müzikleri , karakterleri , müzikleri , karakterleri , müzikleri , karakterleri , müzikleri , karakterleri , müzikleri , karakterleri ile ayrı bir hava katmış inanılmaz . kesinlikle izleyin derim . 5 / 3 . <eos>

23, 4, 3-1
bir film bu kadar mı sıkıcı ve harika olamaz . sadece brad pitt için bile izlenilebilir . <eos>

23, 4, 2-1
ben filmi çok beğendiğimi <unk> beğenmedim ama filmi izlerken çok sıkıldım . bence çok da kötü bir filmdi . sadece brad pitt ' in <unk> sahne gerçekten çok güzeldi . <eos>

23, 4, 1-1
hayatımda izlediğim en kötü filmdi . o kadar akıcı bir film ki hiç bir şekilde <unk> ki , hiçbir şey <unk> bir film değil . <unk> <unk> <eos>

9. The reason for BLEU score difference

Best score is obtained at Epoch 4 and 5. BLEU-1 and BLEU-4 scores for these epochs are added below. These scores are much lower than the ENG-AMAZON dataset used in the reference paper. In addition, when we look at the comments produced, the results are really bad.

The reason why the BLEU score is quite low compared to the ENG-Amazon dataset has a significant effect on the model being stuck in the generation of the token. The lower the score, the higher the severity of this problem. In some cases, almost no comments can be produced.

10. Conclusions

In paper[1], it is stated that they do not receive long comments in this paper because they go into the details of the book. This means that the data source is more restricted and it is also seen that the comments to be produced will be general. Although it is said that comments are produced according to user and product characteristics in the Paper, these comments are produced according to general characteristics rather than these features. Reviews do not give much information about the content of the products.

Considering the performance of the English model, there is no problem with this model that we have written. In order for the performance of the Turkish model to reach an acceptable level, the Turkish data we collect need to be cleaned seriously.

The model has difficulty in long interpretations. Although the LSTM structure is good for long sequential data according to Vanilla RNN, this situation is not valid for a certain length.

The BLEU score may not be a reliable choice to evaluate the performance of our models. For this reason, a new metric can be developed.

The reason why the BLEU score is quite low compared to the ENG-Amazon dataset has a significant effect on the model being stuck in the generation of the token. The lower the score, the higher the severity of this problem. In some cases, almost no comments can be produced.

Accessing the Context vector from any timestep in LSTM with residual links allows the model to learn better by eliminating the need to allocate capacity to remember the context vector in each timestep.

Due to the low quality of the Turkish dataset, as the learning capacity of the model increases, the model is more prone to overfitting.

References

1. Li Dong , et al., “Learning to Generate Product Reviews from Attributes”

2. Gülşen Eryiğit . ITU Turkish NLP Web Service In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014). Gothenburg, Sweden, April 2014

3. http://tools.nlp.itu.edu.tr/