



FINAL NLP CLASS PROJECT REPORT

BBC News Analysis

Mahsa Ghaderan

Supervise by

Dr Sauleh Etemadi

July 17, 2021

Department of Computer Engineering
Iran University of Science and Technology

Contents

List of Figures

List of Tables

List of Acronyms

1	Word2Vec	1
1.1	Iran News	1
1.2	Art News	2
1.3	Sport News	2
1.4	Economic News	2
1.5	Science News	3
2	Tokeniation	5
2.1	Sub-word Level	5
2.2	Word Level	6
3	Language Model	9
3.1	LSTM	9
3.2	GRU	9
3.3	Transformer	12
3.4	Temperature	13
3.5	Final Outputs	13
3.5.1	Iran	13
3.5.2	Art	13
3.5.3	Science	14
3.5.4	Economic	14
3.5.5	Sport	15
4	Finetuning	17

List of Figures

Figure 1.1	Sub-words vocab example	1
Figure 1.2	Sub-words vocab example	2
Figure 1.3	Sub-words vocab example	2
Figure 1.4	Sub-words vocab example	3
Figure 1.5	Sub-words vocab example	3
Figure 2.1	Sub-words vocab example	6
Figure 2.2	Tokenized Corpus	7
Figure 3.1	LSTM Architecture	10
Figure 3.2	Perplexity LSTM language model, label = Science	10
Figure 3.3	Generated Text LSTM language model, label = Science	10
Figure 3.4	GRU Architecture	11
Figure 3.5	Perplexity GRU language model, label = Science	11
Figure 3.6	Generated Text GRU language model, label = Science	12
Figure 3.7	Perplexity Transformer language model, label = Science	12
Figure 3.8	Generated Text Transformer language model, label = Science	12
Figure 3.9	Generated Text Transformer language model, label = Iran	13
Figure 3.10	Generated Text Transformer language model, label = Art	14
Figure 3.11	Generated Text Transformer language model, label = Science	14
Figure 3.12	Generated Text Transformer language model, label = Economic	14
Figure 3.13	Generated Text Transformer language model, label = Sport	15
Figure 4.1	Tokenization based on BERT model for an input sentences.	17

List of Tables

Part 1

Word2Vec

In this part I used assignment2 from cs224n, 2021 Stanford NLP course as my base code. Different Word2Vec models are trained for each label which exist in BBC data set. Headline is concatenated with body of each news for as input text for the model. Each models trains for 40000 iteration. It took about 5 hours for each label. Wights of models are saved is "models/word2vec" directory. Most 30 repeated words are chosen from each label. Images - , - , - and - shows distribution of each model on a 2D map.

1.1 Iran News

After training word2vec model we exoect to see words with same meaning appear near to ech other and words with far meaning and context place far from others on each map.

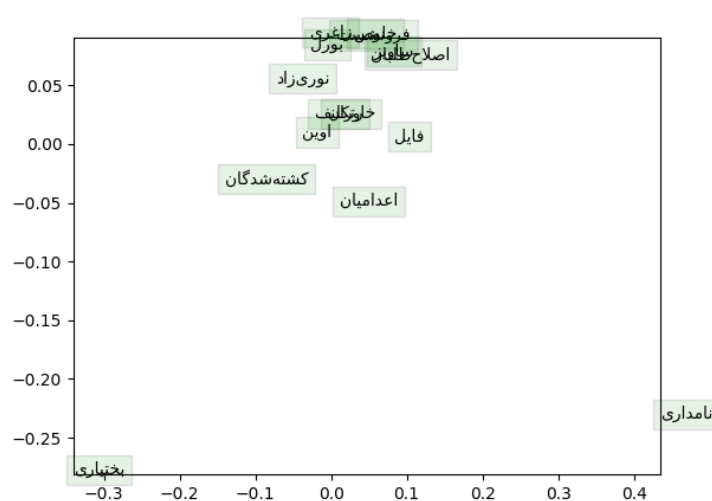


Figure 1.1: Sub-words vocab example

1.2 Art News

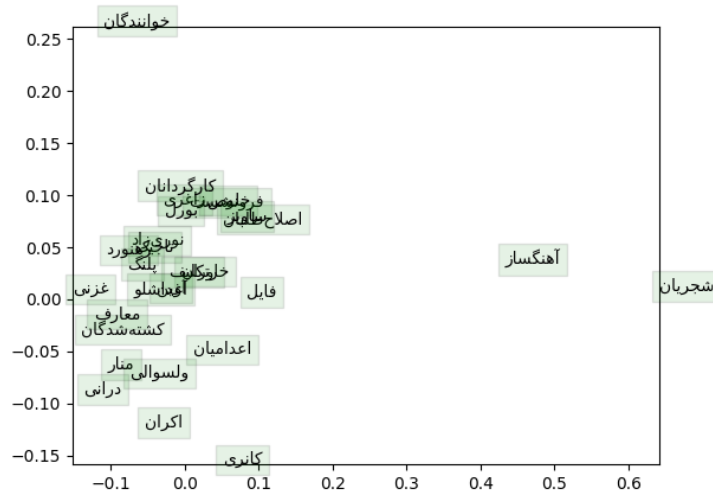


Figure 1.2: Sub-words vocab example

1.3 Sport News

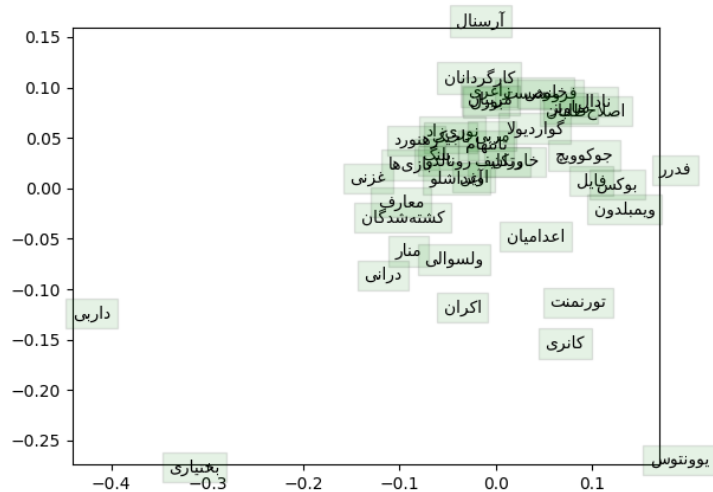


Figure 1.3: Sub-words vocab example

1.4 Economic News

In map 1.4 words in car and it's factory are gathered in the left part of the image.

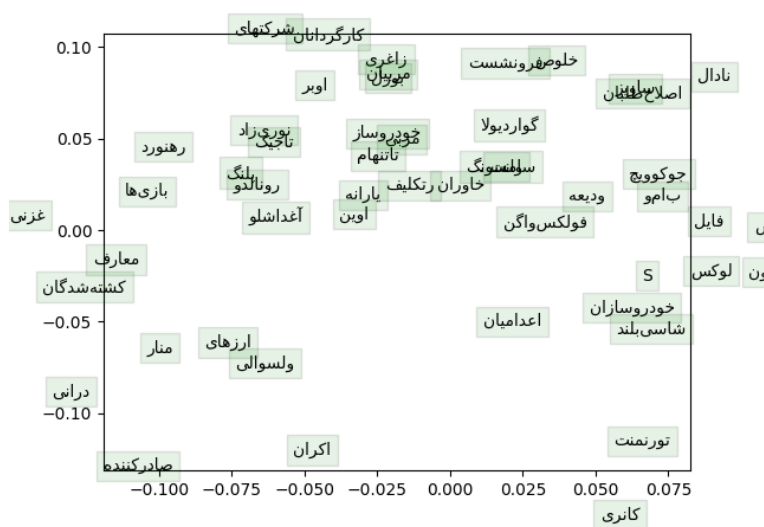


Figure 1.4: Sub-words vocab example

1.5 Science News

At map 1.5 words are overlapped and it's not easy to distinguish them. Equivalent of two words Astronauts and UFO persia are in same context and located almost near to each other and far from others. Two other words are composer and name of a popular Iranian singer, Shajarian, are placed next to each other.

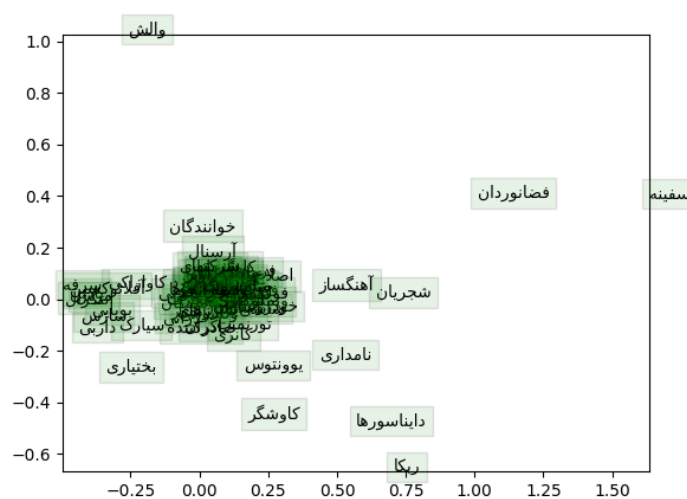


Figure 1.5: Sub-words vocab example

Part 2

Tokeniation

SentencePiece is an unsupervised text tokenizer and detokenizer python project, which mainly used for text generation task. This part is mainly based on this packet. It uses a model to train what tokens are best to tokenizing a given corpus. In this part 2 different methods of tokenization are implemented. First is tokenizing words and sub-words, in this method the model is looking for most repeated words and sub-words and even letter. As a result we can see almost every word in final version of chosen tokens. On the contrary the second method is just tokenizing words, this model is looking for most repeated words.

Until this part entire headline and body of each news are both covered in the corpus but from this part English values are removed from corpus so tokenization and language generation will focus on Persian.

The corpus is shuffled 5 times and separated to train and test data. Also tokenization is applied on 4 different vocab size due to possible SentencePiece size and performance of output model is evaluated against these vocab size. In the following section, we discuss about effects of vocab size from very small value increasing to a large value.

2.1 Sub-word Level

At first I train tokenizer model on sub-words. It means model not only pays attention to words, but also pays attention to sub-words during training time.

When the vocab size is 1000, it is too small for such a corpus and due to model type, model focuses on smaller sub-words rather than meaning full words. In final choosed-vocab, almost every character in source language exists. So when we tokenize a text

there wouldn't be any <unk> token and most words break into sub-words.

As vocab size increases model learns more complicated sub-words and less meaning-less characters and sub-words. But the negative point is as the vocab size increases it would be harder for models to learn task and the positive point is more context will be save after tokenization.

Another point is SentencePiece can discriminate between statrting token and not starting token and ending token of a word in this mode. As we can see in 2.1 the are different form of one token.

```
"<unk>": 3, "ی": 4, "ه": 5, ".": 6, "_": 7,
"ت": 8, "ان": 9, "م": 10, "د": 11, "ل": 12,
"ر": 13, "ش": 14, "است": 15, "ا": 16, "ن": 17,
"می": 18, "ب": 19, "س": 20, "و": 21, "های": 22,
"ز": 23, "ایران": 24, "ی": 25, "ق": 26, "ند": 27,
"ع": 28, "م": 29, "ات": 30, "ور": 31, "ار": 32,
"ف": 33, "ت": 34, "ا": 35, "ن": 36, "خ": 37,
"سال": 38, "ح": 39, "شده": 40, "ج": 41, ",": 42,
"ال": 43, "بر": 44, "یک": 45, "بود": 46, "ست": 47,
```

Figure 2.1: Sub-words vocab example

2.2 Word Level

In this part, i used word level tokenization, in this experience there exist <unk> token due to vocab size and model dealing with choosing best token in order to minimize <unk> token in unseen text.

Same as previous experiment, data-set is divided into train and test set and 5 different shuffled corpus is saved, this help us to have a better evaluation on each vocab size. Four different vocab size is defined as follow: 1000, 5000, 10000, 15000. Number of <unk> token and percent of that over each corpus and vocab is reported in "report/tokenization" directory and each model and output on test set is saved in "src/tokenization/working-dir/words-out". percent of each vocab size is presented in table 2.2.

	word-level					
corpus	1	2	3	4	5	AVG
1000	0.44	0.44	0.44	0.43	0.44	0.44
5000	0.20	0.20	0.20	0.20	0.20	0.21
10000	0.14	0.14	0.14	0.14	0.15	0.15
15000	0.12	0.12	0.12	0.12	0.14	0.13

As it is illustrated in table 2.2, number of <unk> token increase by vocab size reduction. the fewer <unk> token, the fewer information loss. On the other the larger vocab size leads into more amount of computaion for future task. As we saw it is a trade of between these to parameters, in my opinion best vocab size due to this corpus is **10000**. It has reasonable <unk> token and it may need less computation rather than 15000 vocab size.

After tokenizing corpus in word level, we can see <unk> tokens besides tokens which exist in vocabulary. In this project <unk> token id is equal to 3. Many <unk> token can be seen in tokenized validation set (Figure 2.2).

```
.228, 10, 3, 760, 3, 57, 3, 50, 530, 28, 784, 872, 30, 10, 3, 3, 385, 4, 41, 29, 834, 28, 259, 3, 37, 3, 56, 396, 157, 46, 3, 4, 226, 3, 273, 50, 530, 237, 3, 92, /
\ 10, 3, 416, 77, 387, 398, 3, 36, 46, 3, 530, 120, 415, 347, 77, 437, 7, 10, 27, 3, 59, 513, 3, 498, 237, 3, 3, 3], [438, 6, 144, 3, 408, 3, 3, 39, 3, 3, 438, 6, /
.709, 404, 279, 3, 408, 3, 3, 12, 13, 438, 80, 201, 437, 3, 718, 3, 408, 3, 3, 531, 93, 278, 3, 3, 3, 934, 674, 3, 3, 165, 512, 278, 437, 8, 3, 95, 404, 3, 529, /
.408, 3, 408, 3, 3, 665, 114, 3, 408, 3, 731, 3, 37, 149, 3, 20, 147, 3, 823, 3, 3, 3, 8, 471, 3, 37, 360, 509, 3, 37, 278, 130, 3, 901, 636, 29, 3, 3, 48, 94, 943, /
.368, 59, 408, 3, 17, 16, 3, 146, 629, 408, 3, 469, 3, 114, 3, 34, 3, 891, 3, 408, 3, 455, 28, 105, 3, 64, 3, 408, 3, 695, 783, 3, 3, 191, 281, 3, 408, 3, 9, 3, 3, /
.69, 3, 10, 3, 3, 43, 3, 9, 3, 3, 191, 3, 9, 3, 698, 3, 69, 3, 10, 64, 3, 3, 43, 191, 34, 3], [27, 3, 3, 3, 3, 348, 3, 3, 3, 502, 142, 3, 3, 32, 150, 591, 3, 57, /
.3, 3, 3, 3, 233, 3, 461, 24, 308, 3, 3, 3, 150, 9, 560, 211, 3, 19, 3, 111, 3, 3, 168, 3, 3, 3, 303, 3, 3, 19, 67, 22, 883, 6, 3, 654, 3, 58, 213, 349, 3, /
.51, 3, 3, 27, 3, 397, 260, 3, 296, 7, 3, 3, 3], [3, 66, 3, 3, 28, 79, 3, 3, 3, 3, 358, 267, 3, 3, 3, 761, 3, 3, 3, 18, 42, 3, 3, 3, 3, 3, 7, 3, 58, 281, 3], /
.[143, 75, 968, 3, 3, 25, 267, 3, 3, 3, 3, 497, 171, 267, 3, 3, 3, 968, 3, 3, 3, 276, 3, 3, 3, 598, 8, 91, 143, 12, 13, 467, 513, 3, 3, 50, 3, 59, 3, 3, 3], /
.[21, 25, 933, 244, 762, 74, 75, 354, 66, 3, 3, 392, 3, 3, 34, 961, 3, 71, 159, 469, 3, 354, 3, 62, 140, 21, 722, 3, 821, 392, 3, 351, 3, 3, 3, 829, 961, 3, 71, /
.3, 328, 53, 6, 3, 17, 3, 25, 3, 3, 74, 3, 26, 200, 21, 3, 3, 44, 232, 469, 3, 3, 722, 400, 609, 3, 120, 3, 25, 449, 26, 184, 762, 74, 63, 821, 392, 3, 3, 44, 6, /
.273, 3, 354, 3, 722, 400, 21, 402, 3, 3, 69, 78, 3, 3, 3, 762, 886, 64, 191, 362, 3, 3, 217, 762, 74, 34, 12, 13, 996, 636, 29, 44, 6, 34, 30, 232, 354, 3, 722, /
.400, 140, 21, 3, 43, 163, 191, 3, 3, 39, 3], [3, 3, 3, 286, 25, 325, 3, 100, 3, 14, 444, 3, 28, 45, 322, 3, 3, 586, 325, 3, 827, 3, 3, 3, 3, 3, 104, 586, 10, /
.325, 3, 100, 16, 324, 444, 3, 3, 3, 10, 28, 49, 252, 254, 3, 225, 325, 3, 8, 223, 3, 110, 3, 3, 468, 3, 3, 3, 769, 3, 44, 3, 379, 3, 3, 296, 7, 3, 336, 29, 598, /
.114, 36, 16, 67, 47, 3, 3, 55, 224, 53, 3, 3, 19, 335, 85, 252, 3, 444, 996, 3, 103, 3, 886, 3, 225, 325, 596, 3, 216, 3, 3, 411, 20, 595, 3, 446, 325, 529, 39, 3, /
.3, 3, 385, 3, 586, 325, 3, 3, 3, 12, 764, 325, 3, 3, 831, 729, 36, 344, 3, 792, 3, 3, 138, 8, 153, 3, 3, 782, 586, 3, 3, 242, 444, 4, 3, 494, 444, 3, 328, 319, 35, /
.567, 444, 367, 514, 547, 3, 653, 458, 325, 3, 3, 3, 149, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 193, 3, 8, 3], [3, 25, 3, 3, 3, 7, 42, 3, 350, 137, 3, 467, 3, /
.3, 94, 3, 3, 34, 104, 467, 3, 3, 94, 3, 260, 3, 351, 3, 3, 3, 3, 77, 469, 3, 649, 160, 48, 799, 3, 103, 3, 739, 3, 94, 3, 575, 265, 481, 3, 983, 3, 842, 3, 3, 434, /
.224, 3, 104, 3, 3, 3, 3, 262, 3, 3, 3, 20, 178, 3, 77, 42, 3, 3, 3, 235, 348, 3, 3, 522, 585, 3, 36, 100, 13, 3, 3, 27, 4, 3], [397, 564, 51, 129, 66, 23, 534, /
.144, 496, 165, 3, 321, 66, 670, 3, 3, 265, 3, 14, 564, 4, 52, 3, 670, 3, 243, 3, 379, 3, 6, 855, 84, 479, 165, 3, 3, 39, 3, 564, 117, 574, 195, 6, 825, 3, 165, 3, /
.321, 3, 352, 470, 397, 39, 7, 3, 3, 265, 6, 165, 3, 189, 3, 243, 3, 3, 3, 195, 6, 49, 130, 712, 416, 77, 387, 398, 127, 26, 17, 7, 517, 3, 3, 334, 23, 165, 3, 238, /
.3, 661, 3, 517, 189, 3, 104, 243, 46, 76, 3, 243, 3, 379, 3, 6, 855, 84, 479, 165, 3, 3, 39, 3, 3, 3, 55, 372, 3, 3, 3, 3, 321, 855, 36, 7, 103, 59, 27, 3, 59, /
.65, 116, 3, 656, 8, 3, 3, 321, 258, 122, 3, 515, 115, 3, 20, 826, 104, 479, 165, 3, 3, 3, 265, 6, 138, 8, 3, 6, 321, 95, 670, 3, 3, 3, 3, 247, 3, 3, 48, 94, 352, /
```

Figure 2.2: Tokenized Corpus

Part 3

Language Model

Language model neural network architecture is adopted from [pytorch/examples](#) project. Tokenizer codes and models are replaced with Part 2 and vocab for Language Modeling is constructed in part 2. In this part the corpus is divided into three parts, train set, dev set and test set. LanguageModel is trained on each label separately to evaluate capability of each model to generate specific news. Models are saved in "models/lm" directory. Models are trained on GTX1070 GPU. training process is 10 times faster than CPU.

3.1 LSTM

Use LSTM architecture (Figure [3.1](#)) for language model. Best perplexity recorded is 47 on "Science" label. Each epoch lasts about 0.36 seconds, perplexity method starts from 60 and it is 43 on validation data on last epoch.[3.2](#)

In this trial, there are many <unk> tokens in output generated text. But generated sentences are in "Science" context.[3.3](#)

3.2 GRU

Use GRU architecture (Figure [3.4](#)) for language model. Few last epochs and output is illustrated in figure [3.5](#). LSTM and GRU have similar learning power as evidence: perplexity of GRU model on test set is 43, when label is "Science", same as LSTM. Each epoch lasts about 0.33 seconds as expected. GRU model is faster, compared to LSTM model.

In this trial, again there are many <unk> tokens in output generated text. Gener-

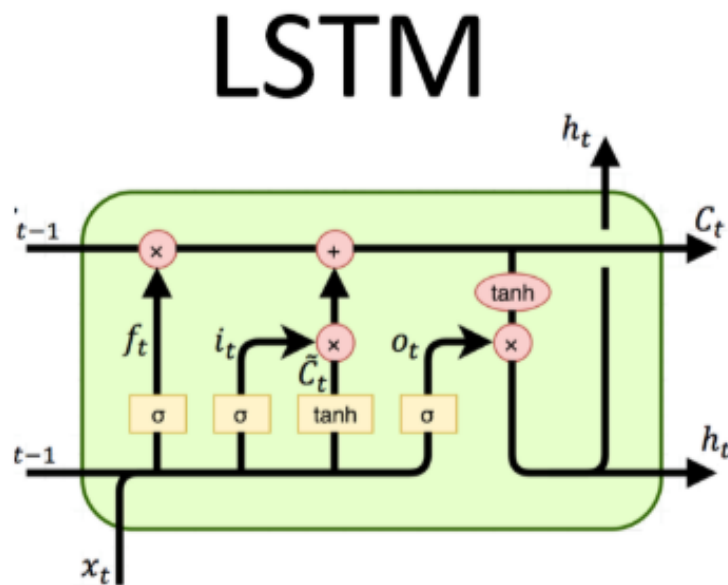


Figure 3.1: LSTM Architecture

```
| end of epoch 35 | time: 0.38s | valid loss 3.86 | valid ppl 47.48
|-----|
| end of epoch 36 | time: 0.36s | valid loss 3.86 | valid ppl 47.47
|-----|
| end of epoch 37 | time: 0.36s | valid loss 3.86 | valid ppl 47.46
|-----|
| end of epoch 38 | time: 0.36s | valid loss 3.86 | valid ppl 47.45
|-----|
| end of epoch 39 | time: 0.36s | valid loss 3.86 | valid ppl 47.42
|-----|
| end of epoch 40 | time: 0.36s | valid loss 3.86 | valid ppl 47.42
|-----|
| End of training | test loss 3.77 | test ppl 43.17
```

Figure 3.2: Perplexity LSTM language model, label = Science

است شکست دیگری هنگام <unk> برگزاری اجرائی بعد <unk> افراد با <unk> سال <unk> <unk> <unk> زمانی کردن داشته همین
_جمعه <unk> <unk> <unk> محیط وزیر <unk> <unk> <unk> خطر دلار چند <unk> <unk> کمک جان <unk> بین ویروس
وجود ماه افراد <unk> <unk> بازی زندگی <unk> <unk> <unk> راه داشته نظیر واکس آبا چند <unk> دو
_فوق <unk> شش ابران خیلی <unk> <unk> <unk> همین صورت <unk> <unk> سال <unk> بود دانشگاه اولین <unk> <unk> هستند،
خواهد ساخته کمی فروردین دو آنها درصد <unk> می است، بیش گونه کشور ممنوع <unk> <unk> <unk> ع
دیگری حال دلار <unk> ویروس زندگی آزمايش بالا <unk> <unk> نقیض <unk> تاریخی <unk> <unk> <unk> اولی <unk>
آزمايش بیش <unk> کره شود، یکی <unk> <unk> <unk> تغییر <unk> تصویر کند کرده موارد نیست <unk> یک
<unk> <unk> تیمم <unk> پزشکی نفر همراه درصد تصویر <unk> بودند بنویس جمع چند چند <unk> <unk> بهبود <unk>
نفر _<unk> <unk> <unk> فاصله مرکز است، بود <unk> <unk> <unk> <unk> تعیین است، دانشگاه <unk>

Figure 3.3: Generated Text LSTM language model, label = Science

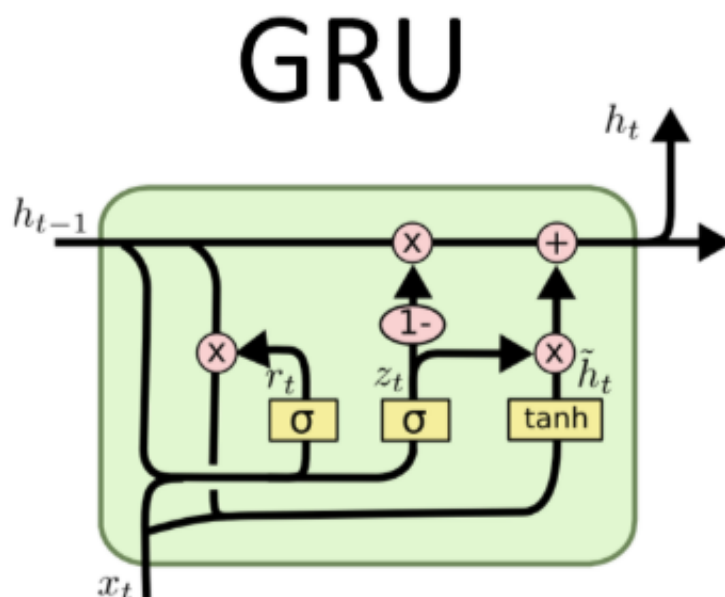


Figure 3.4: GRU Architecture

```

=====
| end of epoch 35 | time: 0.45s | valid loss 3.80 | valid ppl 44.85
=====
| end of epoch 36 | time: 0.32s | valid loss 3.80 | valid ppl 44.81
=====
| end of epoch 37 | time: 0.32s | valid loss 3.80 | valid ppl 44.78
=====
| end of epoch 38 | time: 0.32s | valid loss 3.80 | valid ppl 44.74
=====
| end of epoch 39 | time: 0.34s | valid loss 3.80 | valid ppl 44.68
=====
| end of epoch 40 | time: 0.34s | valid loss 3.80 | valid ppl 44.67
=====
| End of training | test loss 3.69 | test ppl 40.04
=====

```

Figure 3.5: Perplexity GRU language model, label = Science

زمانی کردن داشته خود <unk> <unk> <unk> <unk> <unk> افراد <unk> برگزاری اجرائی بعد <unk> شکست یا هنگام <unk>
فاصله <unk> <unk> <unk> <unk> <unk> وزیر <unk> <unk> <unk> خطر <unk> <unk> هشدار <unk> <unk> کمک جان <unk> بین حاضر
وجود ماه ورود <unk> موسسه <unk> بازی <unk> زندگی <۲۰۲۰> <unk> ده داشته نظیر واکسن آیا چند پیش دو
کرده <unk> شش هفته خیلی <unk> تأثیر صورت <unk> سال مرگ سلامت بهتر دانشگاه اولین <unk> هستند،
خواهد ساخته است. است این دو آنها درصد <unk> بیش دوره کشور ممنوع <unk> می <unk> <unk> <unk> <unk> ع
دیگری حال داری <unk> وپروش زندگی آزمایش بالا <unk> <unk> کرد <unk> هر تاریخی <unk> <unk> <unk> اولی <unk>
آزمایش پیش <unk> کرده شود. یکی <unk> <unk> <unk> تغییر <unk> <unk> تصویر کند کرده موارد با <unk> یک
<unk> <unk> نیمه <unk> همراه درصد حالا <unk> پزشکی <unk> بودن بین <unk> جمع چند گذشته <unk> <unk> بهبود <unk>
نفر ۵ <unk> <unk> <unk> فاصله مرکز <unk> <unk> <unk> <unk> تعیین است. دانشگاه <unk>

Figure 3.6: Generated Text GRU language model, label = Science

```
| end of epoch 37 | time: 0.40s | valid loss 3.74 | valid ppl 42.17
-----
| end of epoch 38 | time: 0.41s | valid loss 3.74 | valid ppl 42.17
-----
| end of epoch 39 | time: 0.41s | valid loss 3.74 | valid ppl 42.17
-----
| end of epoch 40 | time: 0.40s | valid loss 3.74 | valid ppl 42.17
-----
=====
| End of training | test loss 3.62 | test ppl 37.41
=====
```

Figure 3.7: Perplexity Transformer language model, label = Science

ated sentences are in "Education" context.3.6

3.3 Transformer

In last trial Transformer architecture is used for language model. This model construct of a Encoder and Decoder. Encoder consist Embedding layer and Decoder consists Linear layer. Few Last epochs and output is illustrated in figure 3.7. LSTM and GRU both have almost similar learning power, but with transformer architecture it is possible to access better results. As shown in Figure 3.8 test perplexity is 37.41, when label is "Science" which is 3 score better than previous models. Each epoch last longer than previous models and it is about 0.40 second.

In this trial generated text is shown in 3.8. Number of <unk> token is decreased and there are more meaningful sentences. Generated sentences are in "Science" context.

است بر زمین روند <unk> برگزاری اجرائی <unk> <unk> یا <unk> سال <unk> <unk> دارد زمانی می گوید <unk> ۲۰۲۰
 جمعه <unk> <unk> <unk> آب هوایی <unk> <unk> ماه بر سطح فانون <unk> <unk> کمکی جان <unk> دهه بیش
 وجود آمده افراد <unk> <unk> موسسه <unk> بازی می گوید <unk> ۲۰۲۰ <unk> کنار گذاشته نظیر گفته آیا چند ساعت دو
 کشور <unk> شش ماه <unk> <unk> <unk> همین دلیل <unk> <unk> سال <unk> <unk> بخش خبری تلویزیون بخش بودند <unk> <unk> <unk>
 <unk> آنها <unk> زمین ساخته شده است.ایری <unk> کشور فرانسه <unk> می کنند بیش <unk> <unk> <unk> <unk> <unk> <unk>
 داشته <unk> دیگری <unk> <unk> <unk> ویوس کرونا کمکی <unk> <unk> <unk> نقش <unk> هر تاریخی <unk> <unk> <unk> <unk> <unk> <unk>
 آزما بش یک <unk> <unk> کره زمین <unk> <unk> یک <unk> <unk> سال آینده <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 <unk> <unk> نیمه <unk> <unk> تصویری <unk> <unk> همراه <unk> <unk> <unk> پزشکی <unk> <unk> بودند <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 نفر ۵ <unk> <unk> <unk> روند تولید <unk> <unk> دسترسی بوده است تا ثیر <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>

Figure 3.8: Generated Text Transformer language model, label = Science

سخت است. جالی جان خود دست وزیر بهداشت گفته بود گفته <unk> گفت <unk> <unk> <unk> گفته شده <unk> است بر اثر ابتلا بیماری جان خود دست <unk> دست <unk> بازار بزرگ <unk> آغاز شود. هنوز احتمال مرگ جمعه <unk> ایران <unk> وزیر کشور گفته بود <unk> خطر بالای یک <unk> هزار نفر بر روی کار مواد قانون اقدام <unk> موسسه سال جاری اعلام کرد یک <unk> آسیا یک میلیون <unk> آبا حسن روحانی دو کشور <unk> ایران <unk> همین دلیل <unk> شنبه <unk> بر اساس تصمیم <unk> <unk> <unk> بین ۶ هزار میلیارد دلار بول ممونه ستاد اجرایی مجموعه دوره اخیر ممنوع است دست <unk> <unk> فروردین دو روز اول خط فاریس کشور <unk> نزدیک دولت گروه اول مجلس روزنامه ایران <unk> با یک مرکزی تاریخی <unk> <unk> <unk> <unk> <unk> آرما بش <unk> <unk> کره جنوبی آستانه انتخابات ریاست جمهوری ایران. گروه <unk> <unk> تصویر <unk> کرده حسن روحانی رئیس بلیغ ایران است. آقای <unk> <unk> چند هریه <unk> بودند <unk> <unk> پزشکی قانونی استان درصد رسیده <unk> نیمه اول سال داده است. ۵۰ فاصله یک روز روند بررسی <unk> تهران <unk> اعلام کرده است <unk> عالی کشور <unk>

Figure 3.9: Generated Text Transformer language model, label = Iran

Run "bash run.sh 4" to train all three types of language models. Models will save in "models/lm" directory. Then run "bash run.sh 41" in order to generate text for each label. For each model generated text will save in corresponding file in "reports/lm" directory.

3.4 Temperature

Different temperature can be used when generating text. If set temperature low, generated text will be more general, as a result number of <unk> token will increase. On the other hand by setting higher value for temperature model diversity increases. For this project 1 is best choice for temperature parameter.

3.5 Final Outputs

In this section output of each label on trained transformer-based language model is represented.

3.5.1 Iran

Iran news is largest corpus in this dataset. We can see more meaningful sentences and less <unk> tokens. Generated sentences seems link Iran news and generated-words illustrated in Figure 3.9 are related to the context.

3.5.2 Art

Longest n-gram in generated text without any <unk> token is placed on second line, which n is equal to 10(Figure 3.10). This sequence of words is not meaningful enough, but seems fluent.

میزان <unk> بایں <unk> <unk> <unk> <unk> اجتماعى صفحه رسمى است. همه <unk> <unk> <unk> نتیجه <unk> گفت
همه گیری وپروس کرونا مشخص برگزاری جشنواره بعد انقلاب افراد <unk> <unk> تلویزیون <unk> <unk> زمانى قبل شروع خود
جمعه <unk> <unk> <unk> ده سال گذشته <unk> <unk> اش <unk> کننده <unk> جشنواره جان <unk> بین المللى
وجود <unk> برگزاری <unk> امسال دليل چاپ کرد یک <unk> <unk> ده ساله نظیر <unk> آیا چند سال دو
کشور <unk> <unk> برگزار شدنمنبع تصویر <unk> <unk> فیلم <unk> فیلم <unk> زنان بریتانیایی یک <unk> بین
۶ فیلم <unk> <unk> کابل برگزار می تصویر، <unk> چهره ممنوع <unk> خواهد <unk> موسیقی <unk> تصویر، <unk>
<unk> <unk> دیگری <unk> <unk> زندگی <unk> <unk> <unk> تصویر ، <unk> هر تاریخی <unk> <unk> اولی <unk> <unk>
آزمایش <unk> <unk> کره جنوبی <unk> یک <unk> <unk> یک فیلم <unk> تصویر <unk> کرده موارد <unk> <unk> یک
مرد سینما <unk> خبری جمع چند ضربه <unk> <unk> <unk> <unk> نفر کشته شدن حالا <unk> نیمه <unk> <unk>
مقابل <unk> تعیین <unk> تصویر <unk> شهر <unk> <unk> تهران بوده <unk> مرکز <unk> بود. <unk> <unk> <unk> نفر ۵

Figure 3.10: Generated Text Transformer language model, label = Art

نتیجه گیری کرده است. انسان راه دور سال با برزگاری اجرای آب هوایی ماه بر سطح قانون کمک جان دهه پیش وجود آمده افراد موسسه بازی می گوید کشور ۲۰۲۰ کنار گذاشته نظیر گفته آیا چند ساعت دو کشور شش ماه همین دلیل سال بخش خبری تلویزیون بخش بودن کشور زمین ساخته شده است. این کشور فرانسه می کنند پیش از این داشته دیگران ویروس کرونا کمک نقش هر تاریخچه اول آرایش یک کره زمین یک سال آینده اعلام تصویر کرده موارد بیماری یک نیمه تصویر همراه پزشکی بودند کردن ممکن است مزمن بهبودی نفر ۵۰ روند تولیدی دسترسی بوده است تا این تعیین کننده حاضر مبتلا کرونا

Figure 3.11: Generated Text Transformer language model, label = Science

3.5.3 Science

In this section there are many <unk> tokens due lack of data in Science class. Longest sequence of words is 4-grams and used words are more general than other sections.

3.5.4 Economic

In Economic class generated field as it is shown in figure 3.12 most at least 6-grams are construct of meaningful sequences of words and they sounds same as economic news. We can see many 8-grams without any <unk> token in this section.

سیاسی اطلاعات گسترده درباره <unk> نفت <unk> ای ۱۱ بیست ماه مارس کنون بوده بودند مدت بسیاری <funk>
هوایی بر قیمت خواهند رفت شاهد اف <unk> نفت <unk> غرب <unk> <unk> زمانی بسیاری داشته خود
بازار سرمایه <unk> ده سال گذشته میلادی <unk> توافق <unk> دلار هر <unk> <unk> <unk> اعمال <unk> نفت در
ایران؛ گزارش خود <unk> <unk> سال جاری میلادی ۲۰۲۰ <unk> وقت محلی گفته است آبا شیوع کرونا دو
<unk> ایران <unk> تجاری کمتر دو سال پیش پس بحران کشورها خاطر تحريم نفتي آمریکا درباره <unk> افزایش
۶ میلیون بشکه نفت ایران هنوز <unk> نفت بیش ۲۰ درصد افت نرخ <unk> میلیارد دلار در <unk> نسبت رای
دیگری دلار کشور <unk> بانک مرکزی بریتانیا چند سال اخیر اقدامات بانک ایرانی گفته است <unk> اول <unk>
آزمایش کووید ۱۹ کره جنوبی شاهد یک میلیون بشکه نفت کشور اعلام <unk> تصویر <unk> کرده است. بانک سرمایه ایران
گفته اطلاع دولت عربستان سعودی چند هنوز <unk> <unk> <unk> <unk> پزشکی ایران همراه درصد عرضه <unk> نیمه اول سال
افت صادرات نفت <unk> نفتی کاهش نرخ هفت میلیون <unk> احتمال فعال زمینه تجارت نفت گاز عراق گفته <unk>

Figure 3.12: Generated Text Transformer language model, label = Economic

جام_جهانی_فوتبال_ایران_گفت_<unk> تیم_ملی_فوتبال_فرانسه_<unk> <unk> <unk> <unk> <unk> ستاره_با_یرن_<unk>
 همه_<unk> حالا_هنگام_<unk> برگزاری_هر_سال_<unk> <unk> <unk> <unk> <unk> سال_<unk> <unk> زمانی_هنوز_اجازه_حضور
 زبان_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> تیم_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 وجود_آمده_بود_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 کشور_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 ۶_بازیکن_بیشین_فوتبال_ایران_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 کشور_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 آزمایش_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 پس_نتایج_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>
 تهران_بوده_کنار_گذاشته_است_<unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>

Figure 3.13: Generated Text Transformer language model, label = Sport

3.5.5 Sport

In this class, generated text includes of variety of meaningful n-grams without any <unk> token. It is interesting that fluent and meaning 9-grams sequence of words is constructed.(Line 4, Figure 3.13)

Part 4

Finetuning

In this part ParsBert V3 is used as pre-trained Bert model. PyTorch version is used. Due to hardware limitation train batch size is set to 4 and evaluation batch size is set to 8. Model save into reports directory every 500 iteration. In first experiments whole data is used to fine-tune BERT model. As a result BERT is fine-tuned on news data set. Next step is to extract on layer before last one as embedding(a representation for each word) and replace task 4 embedding part with this model. Tokenization result for one sample is represented is Figure 4.1.

ما در هوشواره معتقدیم با اشتغال صحیح دانش و آگاهی، همه افراد میتوانند از ابزارهای هوشمند استفاده کنند. شعار ما هوش مصنوعی برای همه است. xt = "افراد، همه، ##گاهی، آ، و، دانش، صحیح، اشتغال، یا، معتقدیم، ##واره، هوش، در، ما، همه، برای، مصنوعی، هوش، ما، شعار، .، کنند، استفاده، هوشمند، ابزارهای، از، میتوانند، .، است"

Figure 4.1: Tokenization based on BERT model for an input sentences.