Turkish Syllable-Based N-Gram Language Model

EMİRKAN BURAK YILMAZ

Contents

1. Introduction

In this project, the objective was to develop a statistical language model for Turkish using N-grams of Turkish syllables. The steps involved data collection from the Turkish Wikipedia dump, syllabification of words, calculation of 1-Gram, 2-Gram, and 3-Gram tables, applying GT smoothing, calculating perplexity, generating random sentences for each N-Gram model, and analysing the results. All necessary procedures were meticulously executed, yielding conclusive results. For further details, refer to the ***demo.ipynb*** file located within the src folder.

The Turkish Wikipedia dump sourced from Kaggle was pre-processed to ensure uniformity and facilitate subsequent analysis. Initially, the dump file underwent several steps:

1.1. Lowercasing and Character Conversion

- Each line of the dump file was processed to convert all characters to lowercase.
- Turkish characters were systematically replaced with their English equivalents to standardize the text.

1.2. Cleaning and Tokenization

- Non-alphabetic characters were removed to refine the text further.
- An open-source library sourced from Github was employed to segment words into their constituent syllables. This segmentation process facilitated the creation of a new file containing the tokenized representation of the dump.

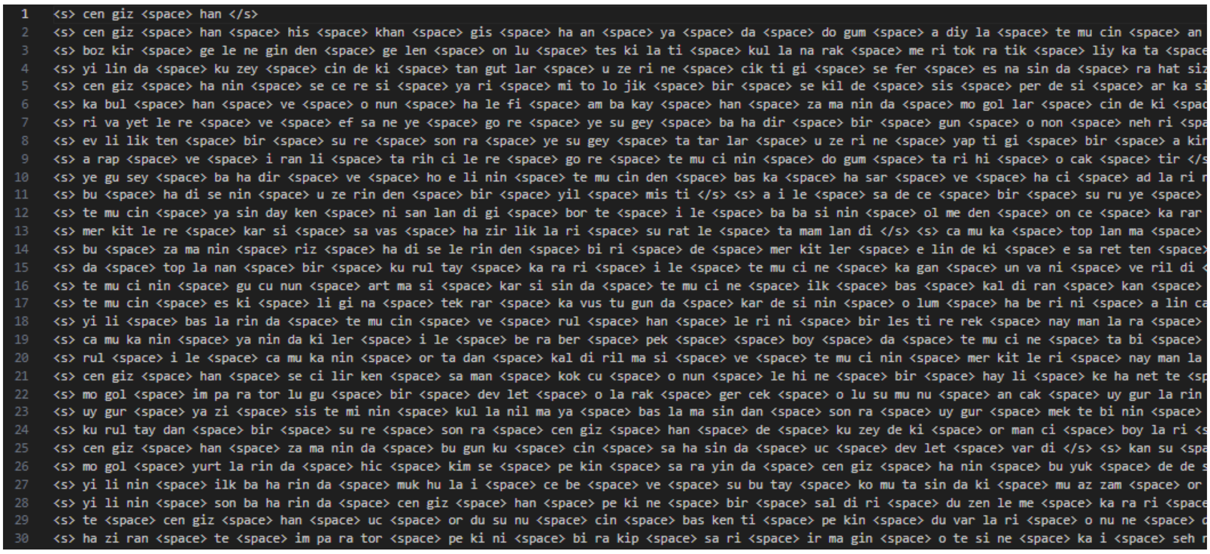
1.3. Tokenized Dump File

- During the tokenization process, care was taken to ensure the preservation of the text structure. Tokens representing the start and end of sentences, as well as space tokens, were included to maintain the structural integrity of the text.

1.4. Splitting into Training and Test Sets

- The tokenized dump was then partitioned into two subsets: 95% for training purposes and 5% for subsequent testing and evaluation of the language model.

By following these steps, the Turkish Wikipedia dump was pre-processed, cleaned, tokenized into syllables for the first non-empty 10,000 lines. The resulting subset was split into appropriate training and test sets for the development and assessment of the N-gram language models.



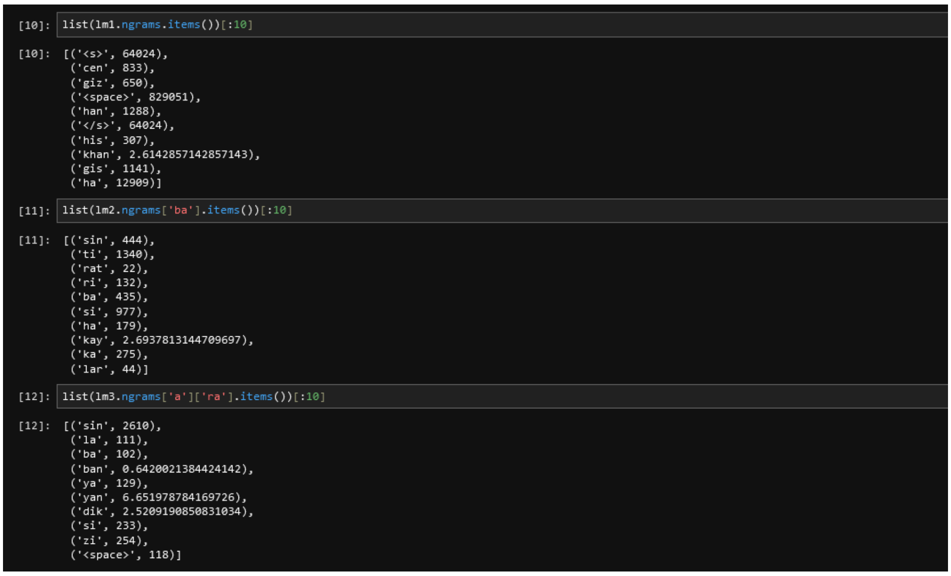
```
<s> cen giz <space> han </s>
<s> cen giz <space> han <space> his <space> khan <space> gis <space> ha an <space> ya <space> da <space> do gum <space> a diy la <space> te mu cin <space> an
<s> boz kir <space> ge le ne gin den <space> ge len <space> on lu <space> tes ki la ti <space> kul la na rak <space> me ri tok ra tik <space> liy ka ta <space
<s> yi lin da <space> ku zey <space> cin de ki <space> tan gut lar <space> u ze ri ne <space> cik ti gi <space> se fer <space> es na sin da <space> ra hat siz
<s> cen giz <space> ha nin <space> se ce re si <space> ya ri <space> mi to lo jik <space> bir <space> se kil de <space> sis <space> per de si <space> ar ka si
<s> ka bul <space> han <space> ve <space> o nun <space> ha le fi <space> am ba kay <space> han <space> za ma nin da <space> mo gol lar <space> cin de ki <spac
<s> ri va yet le re <space> ve <space> ef sa ne ye <space> go re <space> ye su gey <space> ba ha dir <space> bir <space> gun <space> o non <space> neh ri <spa
<s> ev li lik ten <space> bir <space> su re <space> son ra <space> ye su gey <space> ta tar lar <space> u ze ri ne <space> yap ti gi <space> bir <space> a kir
<s> a rap <space> ve <space> i ran li <space> ta rih ci le re <space> go re <space> te mu ci nin <space> do gum <space> ta ri hi <space> o cak <space> tir </s
<s> ye gu sey <space> ba ha dir <space> ve <space> ho e li nin <space> te mu cin den <space> bas ka <space> ha sar <space> ve <space> ha ci <space> ad la ri r
<s> bu <space> ha di se nin <space> u ze rin den <space> bir <space> yil <space> mis ti </s> <s> a i le <space> sa de ce <space> bir <space> su ru ye <space>
<s> te mu cin <space> ya sin day ken <space> ni san lan di gi <space> bor te <space> i le <space> ba ba si nin <space> ol me den <space> on ce <space> ka rar
<s> mer kit le re <space> kar si <space> sa vas <space> ha zir lik la ri <space> su rat le <space> ta mam lan di </s> <s> ca mu ka <space> top lan ma <space>
<s> bu <space> za ma nin <space> riz <space> ha di se le rin den <space> bi ri <space> de <space> mer kit ler <space> e lin de ki <space> e sa ret ten <space>
<s> da <space> top la nan <space> bir <space> ku rul tay <space> ka ra ri <space> i le <space> te mu ci ne <space> ka gan <space> un va ni <space> ve ril di <
<s> te mu ci nin <space> gu cu nun <space> art ma si <space> kar si sin da <space> te mu ci ne <space> ilk <space> bas <space> kal di ran <space> kan <space>
<s> te mu cin <space> es ki <space> li gi na <space> tek rar <space> ka vus tu gun da <space> kar de si nin <space> o lum <space> ha be ri ni <space> a lin ca
<s> yi li <space> bas la rin da <space> te mu cin <space> ve <space> rul <space> han <space> le ri ni <space> bir les ti re rek <space> nay man la ra <space>
<s> ca mu ka nin <space> ya nin da ki ler <space> i le <space> be ra ber <space> pek <space> <space> boy <space> da <space> te mu ci ne <space> ta bi <space>
<s> rul <space> i le <space> ca mu ka nin <space> or ta dan <space> kal di ril ma si <space> ve <space> te mu ci nin <space> mer kit le ri <space> nay man la
<s> cen giz <space> han <space> se ci lir ken <space> sa man <space> kok cu <space> o nun <space> le hi ne <space> bir <space> hay li <space> ke ha net te <sp
<s> mo gol <space> im pa ra tor lu gu <space> bir <space> dev let <space> o la rak <space> ger cek <space> o lu su mu nu <space> an cak <space> uy gur la rin
<s> uy gur <space> ya zi <space> sis te mi nin <space> kul la nil ma ya <space> bas la ma sin dan <space> son ra <space> uy gur <space> mek te bi nin <space>
<s> ku rul tay dan <space> bir <space> su re <space> son ra <space> cen giz <space> han <space> de <space> ku zey de ki <space> or man ci <space> boy la ri <s
<s> cen giz <space> han <space> za ma nin da <space> bu gun ku <space> cin <space> sa ha sin da <space> uc <space> dev let <space> var di </s> <s> kan su <spa
<s> mo gol <space> yurt la rin da <space> hic <space> kim se <space> pe kin <space> sa ra yin da <space> cen giz <space> ha nin <space> bu yuk <space> de de s
<s> yi li nin <space> ilk ba ha rin da <space> muk hu la i <space> ce be <space> ve <space> su bu tay <space> ko mu ta sin da ki <space> mu az zam <space> or
<s> yi li nin <space> son ba ha rin da <space> cen giz <space> han <space> pe ki ne <space> bir <space> sal di ri <space> du zen le me <space> ka ra ri <space
<s> te <space> cen giz <space> han <space> uc <space> or du su nu <space> cin <space> bas ken ti <space> pe kin <space> du var la ri <space> o nu ne <space> c
<s> ha zi ran <space> te <space> im pa ra tor <space> pe ki ni <space> bi ra kip <space> sa ri <space> ir ma gin <space> o te si ne <space> ka i <space> seh r
```

Figure 1: Tokenized Wikipedia Dump

2. Training

The training phase encompassed the development of the unigram, bigram, and trigram language models using the provided training dataset. This involved the fundamental process of counting occurrences and frequencies of respective n-grams within the dataset, establishing their probabilities.

Following the training, an essential technique known as Good-Turing smoothing was applied. This method plays a pivotal role in addressing unseen n-grams by adjusting their probabilities, thereby preventing instances of zero probability. Good-Turing smoothing contributes significantly to enhancing the robustness and generalizability of the language models, accommodating unseen patterns, and ensuring more accurate predictions during subsequent language modelling tasks.



```
[10]: list(lm1.ngrams.items())[:10]

[10]: [('<s>', 64024),
       ('cen', 833),
       ('giz', 650),
       ('<space>', 829051),
       ('han', 1288),
       ('</s>', 64024),
       ('his', 307),
       ('khan', 2.6142857142857143),
       ('gis', 1141),
       ('ha', 12909)]

[11]: list(lm2.ngrams['ba'].items())[:10]

[11]: [('sin', 444),
       ('ti', 1340),
       ('rat', 22),
       ('ri', 132),
       ('ba', 435),
       ('si', 977),
       ('ha', 179),
       ('kay', 2.6937813144709697),
       ('ka', 275),
       ('lar', 44)]

[12]: list(lm3.ngrams['a']['ra'].items())[:10]

[12]: [('sin', 2610),
       ('la', 111),
       ('ba', 102),
       ('ban', 0.6420021384424142),
       ('ya', 129),
       ('yan', 6.651978784169726),
       ('dik', 2.5209190850831034),
       ('si', 233),
       ('zi', 254),
       ('<space>', 118)]
```

Figure 2: n-gram tables were kept as dictionaries

3. Perplexity Evaluation

Perplexity stands as a critical metric in assessing language model performance, reflecting the model's effectiveness in predicting a given text. Lower perplexity values signify superior language modelling capabilities, indicating the model's adeptness at accurately foreseeing unseen data.

Before delving into perplexity calculations, let's examine the probabilities of the sentence.

Sentence	unigram	bigram	trigram
Kablumbağalar uzun yaşar.	1.53569e-32	1.34045e-21	5.23835e-09
Cengiz han dünyaya hükmetti.	4.19830e-29	1.22982e-19	3.70644e-09
Soğuktan üşüyen kediye süt ısıtıp verdi.	5.59456e-47	2.48036e-33	1.77787e-19
Ormanda yürüyüş yaparken, kuş sesleri eşliğinde huzurlu anlar yaşadım. Ağaçların gölgeleri altında dinlenmek gerçekten harikaydı.	6.30688e-131	1.64085e-90	1.19356e-64
Dağların zirvesine tırmanırken, etrafımdaki manzara beni büyüledi. Temiz hava ve dinginlik, hayatımın en güzel anlarından biriydi.	2.41460e-126	1.08329e-97	2.11789e-70

Table 1: sentence probabilities

In this evaluation, the perplexity scores of 1-gram to 3-gram models were computed using the test dataset, constituting the remaining 5% of the total dataset. The test datasets were read line by line and the perplexity of the line were calculated. At the end the sum of perplexities was normalized by the division of number of lines. To prevent potential underflow issues, the logarithm of the chain rule formula's multiplication was utilized in the perplexity calculation process.

Language Model	Perplexity on Test Dataset
unigram	126.94280
bigram	26.63277
trigram	8.58480

Table 2: Evaluation results on test set based on perplexity metric

Sentence	unigram	bigram	trigram
Kablumbağalar uzun yaşar.	280.03171	40.32699	4.33495
Cengiz han dünyaya hükmetti.	152.34930	28.48574	4.45186
Soğuktan üşüyen kediye süt ısıtıp verdi.	159.39962	35.69785	7.81379
Ormanda yürüyüş yaparken, kuş sesleri eşliğinde huzurlu anlar yaşadım. Ağaçların gölgeleri altında dinlenmek gerçekten harikaydı.	136.27910	29.64040	11.16658
Dağların zirvesine tırmanırken, etrafımdaki manzara beni büyüledi. Temiz hava ve dinginlik, hayatımın en güzel anlarından biriydi.	98.61048	34.60437	12.76256

Table 3: n-gram perplexity values on sample sentences

The comparison among language models revealed that the trigram model achieved the lowest perplexity, while the unigram model recorded the highest. This outcome reflects the trigram model's adeptness in capturing contextual information, leading to more accurate predictions for subsequent tokens. A notable trend observed is that as the N-gram order increases, perplexity decreases. This pattern stems from higher-order models having a wider contextual scope, allowing them to capture intricate interdependencies among syllables and thereby enhancing predictive accuracy. Specifically, the trigram model, with its ability to encompass a broader context and dependencies involving three consecutive syllables, is expected to exhibit the lowest perplexity among the tested models. This proficiency enables the trigram model to make more precise predictions, resulting in lower perplexity scores compared to lower-order models.

4. Random Sentence Generation

The most enjoyable part involves crafting random sentences using N-gram models. Following the steps provided in the assignment guidelines, random sentences were generated for each N-gram model by selecting from the top 5 syllables randomly.

Language Model	Generated Sentences
unigram	• ..lalelalelele le la la.lale le .lelela la..la le.la lalalela.lala la.le.la la..lelelala.lelala... lela... la. la. lale.la lela. • . ...lela lalale.le lalalalala .le le. le..lelela lele le. la.lalalalela . ..le lale lalelalalalalala.la. lelala la lelalalale .. • lalale . la..lale lale.lale .. le.lale le le..lale.lalalalale lela.la..la.. lalale .lelalelalale lale. le lela.

	• la .le. .lela ..la le .le.le.. lele la lalalalelele le. la.le.lale le.lala. la..le lalalala. le.lele • lale.le..lele.le le ..le..lalalale ...lela... lala.lelela lale. la .lale lalale.. lelele ..la. lela lalale lelelala.
bigram	• verengibilimcileresinayilindigibiligin olanmalarinayinetirilmek ve ise ve icin onem alamalarinden ola birlerece ozellidir olustur.yinedegininmisti ilereceleri olusmaktaydi bir birlinemindandirmesi • birle olamasinadogu ikisinindadir olusmaktaydi olanmislarlarina anayaziya olustusureket oluslaraktigini verini i alanma verenlerindenlemelerindaginindekiyeti ala isecim onemlerden icindenlerlerinda • iki adi.i icindakiyeni bir birlik alarin ikilinedenilanmayaninmislardagini olusmakta birlidirmeyetiri ve onemlerindahaline birles olusturumuhendi olanmistir. olus birlere olanmistirdigibiligibi • anayilindadirme birlerecesit birligindenlerle olanmistirmelerindendirmasinivermisti birlikleme birlik icindendirma ise birlemeyesi anayaziya alan veri birlestirilmalardandirdi.bulu onemdenilir isekildigiligibi • bulumuha adi.iki adiye veyasayine birlik verinedenizdenlerin birlestir.ala onemler.yinedekiligibi adigibilirler. onemler. verencisi i onemlerden bir i birle verenseltirildi ola i
trigram	• da bulunmuslarlarina karayolun kupalarindaysa verilebilirliklere adina karsisinda ilesini veyahut da adiniminini iceriyordu.yigintinindekilesinegininda.aydinlanma icin.bu onemseyenleriydilerle • rostan anafilenin yapildiktan alan verilerekta bulu ana gorecelinince isein verenlererasinaviniminininluteryenler.adalarin yasasi verileri bulunabilgisa dayanirlar tarihli veya sahipliginagini a • tonda bulunan.o anadoludakilesimli araciligine.ayri ve onem icinden bilimci birlikinsa bir olaraksama ya daginin onemliydi aracinin enzimdir.adasidirler.ozellesmeyeceklerdedirde olacak. • tepe veri olusturmasinabilimlererasinabilimin bir yapilan verenle ikilestirilendirmeye verilir.ocak.yiginin.yilininінluteryen takma olusmayacaksa verildikle illeriyken birlikin ozelligiyla alandaki • mercilerinedirmele gerek adige icerikler onem ikisiliklardalerde.yiginin.ayninacaklarincasinavi olasilik alarak adige uzerilerdedirler icerisindekinele gelerin analitik aracinin.yilindan

Table 4: Generated random sentences

Generating random sentences from N-gram models involves a process where syllables are selected based on their probabilities within the trained model. The trigram model, due to its advanced contextual understanding, is expected to create sentences that are more coherent and contextually relevant compared to lower-order models. Higher-order N-gram models typically generate sentences with better grammar and semantic connections because they can grasp longer dependencies within the dataset. The sentences produced by the trigram model are anticipated to display superior coherence, offering more contextual and grammatical sense than those from lower-order models. Conversely, as the N-gram order decreases, the sentence quality may decline, resulting in less coherence, nonsensical phrases, or grammatical errors. This decline happens mainly because lower-order models have reduced capability to capture intricate linguistic relationships and contextual subtleties found in the dataset.

Note: The used Wikipedia subset have a big impact on the generated random sentences.

5. Conclusion

In summary, the evaluation of perplexity and random sentence generation supports the expectation that higher-order N-gram models, particularly the trigram model, perform better in terms of perplexity scores and the coherence of generated sentences. These findings align with the general understanding that models with more context and higher-order dependencies tend to yield more accurate predictions and produce more coherent text.