N-GRAM LANGUAGE MODELS

ASSIGNMENT 1

SADIK ADEM EKİCİ
ademekici90@gmail.com

March 21, 2019

Introduction

Language models are models which assign probabilities to a sentence or a sequence of words or, probability of an upcoming word given previous set of words. Language models are used in fields such as speech recognition, spelling correction, machine translation etc.

An **n-gram** is a contiguous sequence of n items from a given sequence of text. Given a sentence, s, we can construct a list of n-grams from s by finding pairs of words that occur next to each other. For example, given the sentence “I am Sam” you can construct bigrams (n-grams of length 2) by finding consecutive pairs of words.

PART 1: Building Language Models

In this section unigram, bigram and trigram language models were created. During the creation of the language model, the Federalist Papers, which were documented in writing, were read separately and author specific language models were created. After reading the articles, the sign indicating the end of the sentence was determined.The remaining punctuation marks have been removed. This is because the remaining punctuation does not reflect the identity of the author. Also the words in all essays have been converted to lowercase. 3 different language models will be explained under different titles.

Unigram Models

While the Unigram model was created, there was not much action after the operations described above. After this process, any author found out word count he used in all his articles. And then a special unigram language model to author was created.

Bigram Models

The number of consecutive word pairs were determined when creating the Bigram language model. In addition, when creating the bi-gram language model, the sentence is started with which word, the sentence is ended with which punctuation and which word, these are important case. For this reason, the word "<s>" was added to the beginning of the sentence. The word "token" was added to the end of the sentence. And then a special bigram language model to author was created.

Trigram Models

When creating the Bigram language model, the number of three consecutive words was determined. As in the bigram model, in the trigram model; When creating the trigram language model, the sentence is started with which word, the sentence is ended with which punctuation and which word, these are important case. Unlike the bigram model "<s> <s>" per sentence and the end of the sentence "token token" word has been added. And then a special trigram language model to author was created.

Conclusion

Authors’ language models show the author’s identity. In the following sections, author unknown articles, the author will be estimated. This will be done according to the model of the author we created in the prediction. And the essay will be derived from the language of an author. This will also be done by using the models we created. Of course, the most robust prediction would be for trigram models, because it looks at triple word groups, and this actually reveals more of the author's identity.

PART 2: Automatically Generating Essays

In this part, after the creation of language models in the previous part, the article will be written for each author using each model (unigram, bigram, trigram). And these articles will be tested later. When creating an automatic essay, the words from the model of the cumulative probability generated from the author's model are randomly selected. The models we creation have advantages over each other. These will be discussed in the test part of this part.The evaluations of these tests for unigram, bigram and trigram will be examined under separate headings.

Unigram Models Generator And Test
A single cumulative probability scala is enough to generate an article from the Unigram model. So all the words are put a cumulative probability scala and randomly selected someone.

When the auto-generated articles were tested according to the unigram model, the authors' perplexities were very close to each other. The articles created automatically from the unigram model consist of meaningless and independent words, so the articles created cannot reveal the identity of the author. The Unigram model lags behind other models for author identification and automatic article production. E.g; When the Hamilton model article is tested, Madison and Hamilton's perplexity values are very close to each other.
Here are some articles and perplexities for the unigram model;

****

MADISON UNIGRAM ESSAY: have residuary of the same considered the us they in parts plan collective not as is prosperity essential were actuated the the danger to the effect every however the.

HAMILTON: 339.5350590728663 MADISON: 306.9885645433899
****

HAMILTON UNIGRAM ESSAY: would not to possible that vigor to the the even general war extent upon to improprieties impracticable of not for may last of to a neighbors.

HAMILTON: 168.73979851060193 MADISON: 228.5758097902171
****

Although the examples are derived from their own models, the values of the two authors' perplexities are very close.Therefore, the authors can not show their identity in this model.

Bigram Models Generator And Test
In this model we should not hold a single cumulative probability scale as in the unigram model. Because; bigram moves every word selection depending on the previous word. That is, each time a new word is added, we need to create a cumulative scale according to the new word added. An article created in this way is more meaningful than the bigram model. Because they are composed of two connected word groups. According to the Unigram model, the difference between perplexities is more decisive. So the identity of the author appears more. Here are some articles and perplexities for the bigram model;

****

MADISON BIGRAM ESSAY: the discord and may not to the rights within the enumeration or obstruct the organization of ambition or obstruct the former letters of these principles in the public improvements.

HAMILTON: 4980.762539017282 MADISON: 2637.8028859106066
****

HAMILTON BIGRAM ESSAY: the hands of recurring to exceed the union should not stand ready to apprehend that this is the evidence along through the confederacy.

HAMILTON: 3151.1195466780678 MADISON: 5152.275528812399
****

As seen in the test results, perplexity values of the authors in the bigram model are more distant from each other. Therefore, the bigram model is more decisive than the unigram model.

Trigram Models Generator And Test
As in the Bigram model, the single cumulative probability scale is not sufficient in this model. Because; trigram searches for words that might come after the binary word group. For this model, a new cumulative probability scale is created for each binary word. For this reason, more meaningful sentences occur compared to the other two models. In this model, author identities are more pronounced than other models.
Here are some articles and perplexities for the trigram model;

****

MADISON TRIGRAM ESSAY: it is probable that a bill of attainder ex post facto laws and different tribunals of justice with all their local influence in effecting a change of federal officers.

HAMILTON 22157.526178402855 MADISON 10323.277010559972
****

HAMILTON TRIGRAM ESSAY: the facts that support this opinion are no longer objects of war and this still more powerful combination nor do there appear to harmonize in this respect to the.

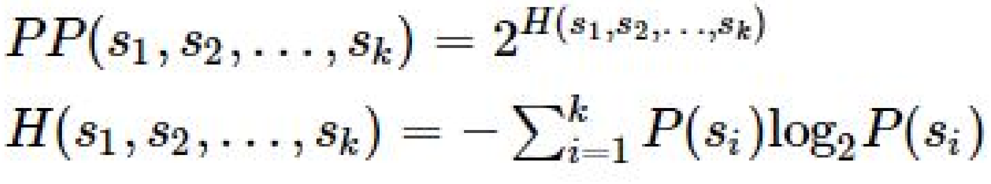
HAMILTON 11719.132551745473 MADISON 21739.194877502145
**As seen in the test results, the authors' astonishment values in the trigram model are more distant than the other models. Therefore, the trigram model is more decisive than other models.

Conclusion
In this part we learned how n-gram models work and what they do. As a result, n-gram language model n number of more accurate estimates as we have created more meaningful sentences. That is to say, among these three models, trigram is superior to other models and the author has revealed its identity more. We have proved this with the tests.

PART 3: Classification and Evaluation

The language models obtained in this part will be tested. In Dataset, the essay of unknown author is available. The authors of these articles will be estimated using the author models created. **Perplexity** values of the models will be used when making this prediction. As the language models created during the perplexity calculations are not sufficient, some words in the test articles may not be in the models. This leads to a probability of 0. To solve this problem, **add-one (Laplace) smoothing method** was used.

What is Perplexity?
In natural language processing, perplexity is a way of evaluating language models. A language model is a probability distribution over entire texts. N-gram language modeling is the most basic and common statistical language modeling method used.
Perplexity of a text is defined in terms of the entropy of its sentences (H). The entropy, in turn, is defined in terms of the probability distribution defined by the language model.



$$PP(s_1, s_2, \ldots, s_k) = 2^{H(s_1, s_2, \ldots, s_k)}$$

$$H(s_1, s_2, \ldots, s_k) = -\sum_{i=1}^{k} P(s_i)\log_2 P(s_i)$$

A low perplexity indicates the probability distribution is good at predicting the sample.

What is Laplace Smoothing?
In the context of NLP, the idea behind Laplacian smoothing is shifting some probability from seen words to unseen words. In other words, assigning unseen words some probability of occurring.
The probability calculation of unseen words is done as follows;

$$P(w_s) = \frac{C(w_s)+1}{N+V}$$

Conclusion Results

The results are not consistent in each model. The Unigram model found the right author in all tests, but sometimes the bigram and trigram models didn't get the right result. The reason was that the dataset we had was insufficient. N-gram models need more data as the N number increases. For this reason, some models do not work properly because there are not enough articles. Normally the trigram model should have better writer detection. Because, according to the models used in the trigram model is expected to reveal the identity of the author more. Once enough data is obtained, the trigram model will work more consistently.

The models did not work properly in the only 9th article. In the test of the 9th article, the bigram and trigram model gave incorrect results. In all the other models, the models are getting correct results.

In general, the winner of the test of articles by unknown author was Madison. This may be because Madison uses a lot of punctuation. Because the dataset is insufficient because of this, punctuation marks have a great effect on probability. In these articles, punctuation marks were used a lot, so Madison was superior.