CS4442 – Assignment 2

Dustin Dobransky 250575030

1. This question required us to compare n-grams between two texts.
   1. For Dostoyevsky’s works, the results were as follows:

|  |  |
| --- | --- |
| nGram size | Percentage of works in 2 not in 1 |
| 1 | 40.647 |
| 2 | 72.681 |
| 3 | 89.796 |
| 4 | 97.606 |
| 5 | 99.536 |
| 6 | 99.891 |

The largest common n-gram occurs at n = 17, with “repulsion that s what i m afraid of that s what may be too much for me”

* 1. For Dickens vs. Kafka’s works, the results were as follows:

|  |  |
| --- | --- |
| nGram size | Percentage of works in 2 not in 1 |
| 1 | 32.880 |
| 2 | 77.458 |
| 3 | 95.433 |
| 4 | 99.195 |
| 5 | 99.885 |
| 6 | 99.977 |

The largest common n-gram occurs at n = 7 with two n-grams:

1. “in the middle of the table and”
2. “there is no such thing as a”
   1. For “MarxEngelsManifest” vs. “SmithWealthNations”, the results were:

|  |  |
| --- | --- |
| nGram size | Percentage of works in 2 not in 1 |
| 1 | 84.185 |
| 2 | 97.468 |
| 3 | 99.534 |
| 4 | 99.920 |
| 5 | 99.987 |
| 6 | 99.998 |

The largest common n-grams occur at n = 6 with:

1. of nature and of reason the
2. is the same as that of
3. to keep up the rate of
4. in order to keep up the
5. of a man s own labour
6. from them what they have not
   1. Discussion:

From the results in a), it can be seen that books written from the same author will have a large number of similar n-grams. This is expected, as the author has a distinct writing style, which will be reflected in their works. In this case, similarity in writing style is identified by the recurrence of n-grams across different novels.

For the remaining two tests, both were authored by different people, and the differentiation between n-grams was large. Past 6-grams, the works did not contain significant overlap.

1. This problem involved determining the percentage of 0-probability sentenced in a test document given a training document. This was done by constructing a hash set of n-grams from the first document, and then generating a hash set for each sentence in the second document. Any of the n-grams from a sentence from the second document was not found to be an n-gram from the first document, then the probability of the sentence existing in the first document was 0. The results from these tests were as follows:

|  |  |
| --- | --- |
| n-gram size | 0-probability sentences [%] |
| 1 | 85.878 |
| 2 | 96.756 |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |

1. This problem involved estimating sentences using a given text. Ngrams ranging from size 1 to n were generated from the text, and the probabilities of generating a next word from the entire vocabulary.
   1. For n = 1,2,3,4,6 the results were as follows:

|  |  |
| --- | --- |
| **n-gram size** | **Sentence** |
| 1 | the begun s you the was and <END> |
| 2 | the cover around his trial at k <END> |
| 3 | the cover of one who can go with you <END> |
| 4 | the cover of one of the people up on the window sill and went to the front entrance waited there in ambush and every time a lawyer tried to enter the building he would throw him down the steps <END> |
| 6 | the cover of one of them had nearly broken through in its middle and it was held together with a few threads <END> |

For n = 1, the results were gibberish. Without any context, the next generated word was just the next most frequent word. With n = 6, there was only one instance of the previous sentence existing, so only one word in the vocabulary had a probability of 1, and the rest 0. This resulted in an exact sentence from the text being generated.

* 1. For n = 3, running the program on MarxEngelsManifest, the result was:

“of nations and crusades <END>”

With little context, a 3-gram size was sufficient to generate a sentence that was actually found in MarxEngelsManifest.txt. This suggests that as more data is provided to the algorithm, larger n-grams can be used, and therefore more context can be generated, leading to better sentence generation.

1. Not complete
2. In this problem, sentences from various languages were tested against language models for 6 different languages, and classified with the most probable language using the Add-Delta classification method. The following is the resulting error for the various test cases:
3. No delta

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 1 0 50 | 14.34 |
| P5 2 0 50 | 19.12 |
| P5 3 0 50 | 47.25 |

1. Delta = 0.05

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 1 0.05 50 | 13.07 |
| P5 2 0.05 50 | 1.04 |
| P5 3 0.05 50 | 0.56 |

1. Varying Delta

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 3 0.05 50 | 0.56 |
| P5 3 0.005 50 | 0.48 |
| P5 3 0.0005 50 | 0.64 |

1. Discussion:

With only ML classification, as seen in b), error increases drastically with larger n-gram sizes. With the addition of Add-Delta smoothing, the error significantly decreases. This is because add-delta takes into account unseen n-grams, which evens out the probability of an n-gram occurring in a particular language.

1. The following is the error found while using varying sentence lengths to determine the language that sentence belongs to.

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 2 0.05 10 | 22.86 |
| P5 2 0.05 50 | 1.04 |
| P5 2 0.05 100 | 0.16 |

The reason why the error decreases with greater sentence length is because there is more context available per sentence. With a larger amount of context, more n-grams will match with the n-grams from a particular language, leading to greater probability of the language being estimated correctly.

1. The following section repeats the tests from b-d, but uses only latin characters, and a vocabulary size of 26. The results are as follows:
   * 1. No delta

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 1 0 50 | 69.64 |
| P5 2 0 50 | 68.61 |
| P5 3 0 50 | 69.08 |

* + 1. Delta = 0.05

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 1 0.05 50 | 65.66 |
| P5 2 0.05 50 | 19.04 |
| P5 3 0.05 50 | 7.41 |

* + 1. Varying Delta

|  |  |
| --- | --- |
| Program Parameters | Error [%] |
| P5 3 0.05 50 | 7.41 |
| P5 3 0.005 50 | 5.42 |
| P5 3 0.0005 50 | 4.54 |

The performance during this round of tests was significantly poorer than the previous round. Since a smaller vocabulary size was used, and all characters were lower case (latin\_only = true), there was a smaller set of data for the Add-Delta language model algorithm to operate on. By reducing the volume of data available, the algorithm was less accurate. Context which could have been important in determining the language of a sentence would have been removed by making all letters lower case.