Name1

Name 2

Name 3

ΗΥ543 – Project Proposal

# Introduction

We will try to find the next word in a input sentence of 4 words thus our machine learning algorithm will predict the 5th word based on a 5-gram dataset. Our algorithm’s prediction will be the most common word that follows the 4-words given sentence.

**from Wikipedia:** an n-gram is a contiguous sequence of n items from a given sequence of text or speech.

Generally, n-grams are fixed size tuples of items which can be syllables, letters or words. In our case the items are words extracted from the Google Books corpus, so they follow a syntactically correct sequence. The n specifies the number of elements in the tuple, so a 5-gram contains five words or characters.

For more info about the dataset, please visit <https://aws.amazon.com/datasets/google-books-ngrams>

This seems, as we said earlier, that can be expressed as a n-gram and as such falls under the category of the language model. A language model is basically a probability distribution over a sequence of words so a possible task would be to assign a probability to each of the words. This seem like a solid approach so we will discuss it later on and also state the drawbacks.

The main reason that we cannot beat the “state of the art” is that we aim at a simple probability model with explicit weights(we will explain later) and also because our dataset is not fit well enough for the particular task(it consists of literature, not natural speech).

# How big is the Dataset

-**English One Million**

(src: <http://storage.googleapis.com/books/ngrams/books/datasetsv2.html> )

|  |  |  |
| --- | --- | --- |
| **Data** | **Rows** | **Compressed Size** |
| 1-gram | 261,823,186 | 2.6 GB |
| 2-gram | 3,383,379,445 | 32.1 GB |
| 3-gram | 10,565,828,499 | 94.8 GB |
| 4-gram | 12,987,703,773 | 113.1 GB |
| **5-gram** | **8,747,884,729** | **75.8 GB** |

As we already mentioned, we are going to use the 5-gram dataset, that contains almost 9 billionrows and its compressed size is about 76GB. The dataset is split into multiple files.

For selected the first file whose uncompressed size is 340MB(compressed size is about 40MB). Since using the whole uncompressed dataset would be way too large (about 650GB) we will beusing about 200GB, 1/3 of the size, so the estimated number of rows would be about 3 billion.

\*\*the chosen size of 200GB is based on the fact that our team consists of 3 members so we havemore amazon sources than the regular 2-person team.

# Description of the Dataset

Each row is a tab separated value containing the following fields:

**5-gram** - The actual 5-gram(5 first values)

**year** - The year for this aggregation

**occurrences** - The number of times this n-gram appeared in this year

**pages** - The number of pages this n-gram appeared on in this year

**books** - The number of books this n-gram appeared in during this year

e.g.(dataset format) :

How Fares my Juliet 1595 1 1 1

**Dataset bias**

Since we are going with a probabilistic approach and the dataset was generated by a deterministic algorithm (5-grams were produced by reading occurences in books), we do not expect any bias with the data. The only bias would lie in the selection of the books, but since their range is from 1520 to 2008 we tend to believe the selection is objective

# Pipeline Stages

We will follow the below process:

As a first step we need to filter the dataset in order to remove whitespaces and special characters, since these are not needed. We are going to use the 5th term of the 5-gram as a label, since the user input will be 4 terms and it is expected to predict the 5th. As a result, the first 4 terms will be used as features, as well as the occurences of each 5-gram (and probably the year of appearance). As a mathematical formulation we will be using explicit weights on each feature to create the baseline model ( e.g. the 4th term will have a greater weight than the 3rd), but the exact weights have not be determined yet. The problem that occurs is that the weights may not be optimal in our first approach, but we are planning to optimize each weight with the use of the validation set, at the tuning stage.

# Evaluation Metric

We mentioned in the introduction about the difficulties that we are facing with probability distribution. Our main concern is that a common evaluation metric for a language model algorithm is the perplexity:

in information theory, **perplexity** is a measurement of how well a probability distribution predicts a sample.(more info: <http://www.itl.nist.gov/iad/mig/publications/proceedings/darpa98/html/lm30/lm30.htm> ) but the mathematical formula for computing perplexity often requires some data(e.g. user history for auto fill suggestions) that our dataset does not provide.

# First 5 lines of dataset

These are the first 5 lines of the dataset, which also indicates why the dataset needs filtering.

"! "" ""He got" 1846 1 1 1

"! "" "" He got" 1855 1 1 1

"! "" "" He got" 1856 1 1 1

"! "" "" He got" 1857 1 1 1

"! "" "" He got" 1859 1 1 1

# Cluster usage

Since the dataset will be about 200GB we will use about 15 M4 instances of type xlarge (4x CPU, 16GB RAM, 750 Mbps disk BW, $0.215/hour).

This will cost about $3.2/hour for each account. This results to about 9 hours per user, so combined we have about 27 hours of the cluster usage.

Since we will be working mostly locally on a smaller portion of the dataset, we believe that for testing purposes these hours will be enough to complete the testing phase and that we will reach to conclusions without exceeding our credits.