Niv Yaakov- 315311118

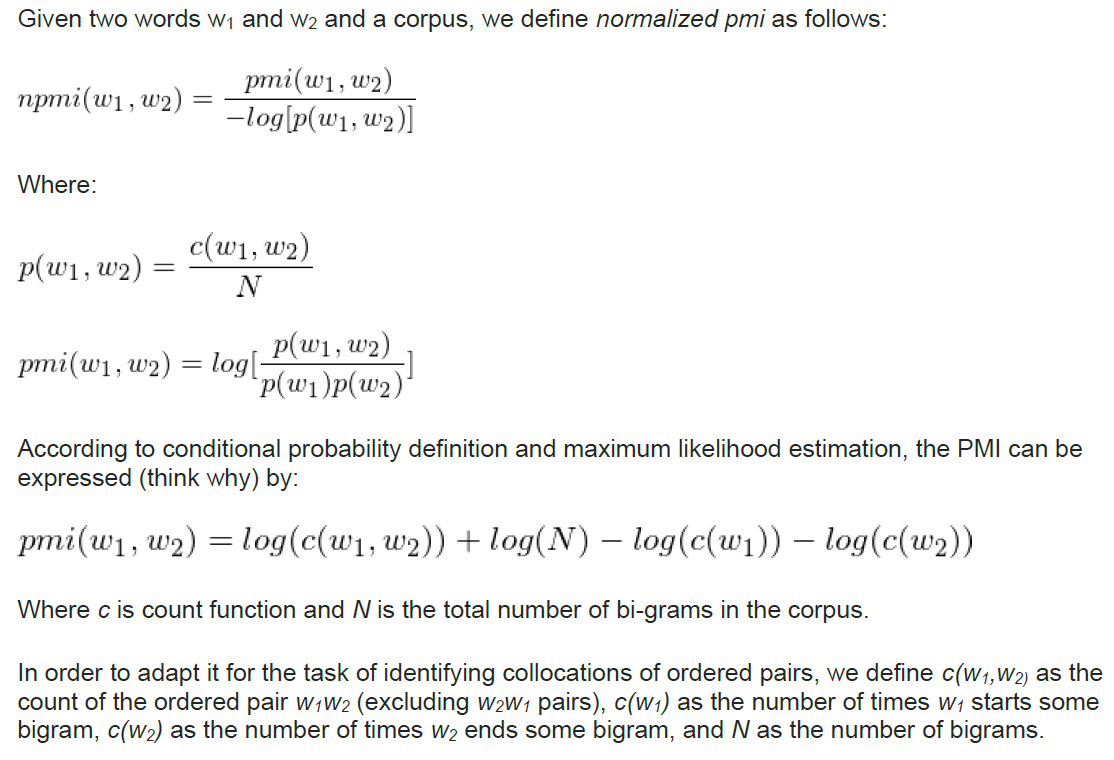
Nitay Maoz- 313359341

Application that automatically extracts collocations from the Google 2-grams dataset using Amazon Elastic Map Reduce

Stack: AWS Java SDK, Hadoop, EMR, S3

A collocation is a sequence of words that co-occur more often than would be expected by chance. The identification of collocations - such as 'crystal clear', 'cosmetic surgery' - is essential for many natural language processing and information extraction applications.

We will use Normalized Pointwise Mutual Information (NPMI), in order to decide whether a given pair of ordered words is a collocation, where two ordered words with a high NPMI value are expected to be a collocation.



Map-Reduce program which produces files of collocations for each decade (1990-1999, 2000-2009, etc.).

Input: Google Bigrams  
▪ Hebrew: s3://datasets.elasticmapreduce/ngrams/books/20090715/heb-all/2gram/data

Algorithm:  
We are looking for 4 parameters to complete the equation mentioned above:  
N: all words from the corpus[for each decade], we count them when inputting the bigrams in the first step Map  
cw1: number of occurrences of every 1st word in a bigram per decade.  
cw2: number of occurrences of every 2nd word in a bigram per decade.  
cw1cw2: number of occurrences of the bigram itself for each decade.

Our algorithm composed of three Hadoop jobs:

**First job:**

receives the Google 2 grams data and each mapper gets a line from the data.

**Mappers input format[for each line]-** w1 w2 /t year /t count

**Mappers output[also the Reducer input]-** 2 kinds of output:

**key**: w1,\*,decade  **value**: count

**key:** w1,w2,decade **value**: count

also calculates N for each decade using Hadoop global counters.

**Reducer output-**

**Key**: w1,w2,decade **value**: cw1w2@cw1

**Second job:**

**Now we will calculate cw2**

**Mappers input[for each line in the first job output]:** w1,w2,decade /t cw1w2@cw1

**Mappers output[also the Reducer input]-** 2 kinds of output:

**Key:** w2,\*,decade **value:** cw1w2

**Key:** w2,w1,decade,cw1 **value:** cw1w2

**Reducer output-**

**Key:** w1,w2,decade,cw1,cw2 **value:** cw1w2

**Third job:**

Now we will calculate the npmi for each w1,w2 for each decade.

**Mappers input[for each line in the second job output]:**

w1,w2,decade,cw1,cw2 /tcw1w2

**Mappers output[also the Reducer input]-**

**Key:** npmi w1 w2 decade

**Reducer output-**

Prints the output to the files[each file represents a decade]-

decade w1 w2 npmi

**How to run-**

1. Define your .aws directory in user.home with a text file called credentials with the relevant credentials.
2. create a bucket with 3 directories inside for the jar,logs and output.
3. Change the s3 path in line 55 in class LocalApp to your relevant path to the jar of the project.
4. Change the s3 path in line 75 in class LocalApp to your relevant path to the s3[the logs will be uploaded here.]
5. Change the s3 path in line 19 in class StepsRunner to your relevant path to the s3[the output will be uploaded here.]
6. Create a jar with StepsRunner as the main class and upload it to the s3 path of your jar.
7. Run the LocalApp class on your computer.