**▪ Names and ID's:**

- Guy Turevich 208466920

- Matan Naftaly 315578641

**▪ Instructions on how to run the project:**

1. Update the credentials file on your local computer at the following address:

- Windows: C:\Users\USERNAME\.aws\credentials

- Linux: ~/.aws/credentials

2. Change the name of the bucket on S3 to the name of your bucket (Line 33), and uncomment Line 29 to create a bucket if it was not yet created.

3. Optional: you can change the value of DPMin in line 38 of Step1 (we used 4).

4. Run the main of the LocalApp class (make sure the boolean variable "local" is set to false in Step1 and Step2).

5. After getting the output of Step2, run the python Classifier on it.

**▪ Design**

The components of the system and their functionality - input and output:

**LocalApp:** Connecting to the AWS, uploading the steps JARs to S3, and starting the steps of the map reduce job.

**Classifier:** Using the Scikit-Learn library to train the model and test it, calculate the precision, recall and f1-measure, and print examples of noun-pairs for each case (TN,TP,FP,FN).

**Dependency Graph:** Models the dependency graph of an input sentence and generating patterns of it.

Fields:

graph- Directed graph representing a sentence.

count = number of appearances in the corpus.

patterns = a list of all the patterns and their count.

**Node:** Models a single vertex in the dependency graph.

Fields:

Word: a word in the sentence

posTag: a Penn-Treebank part-of-speech tag

depLabel: a stanford-basic-dependencies label.

headIndex: an integer, pointing to the head of the current token. “1” refers to the first token in the list, 2 the second, and 0 indicates that the head is the root of the fragment.

**NounPair:** Two words from the corpus, their count, and is word2 is Hypernym of word1 (initially set to false, determined later).

**Step1:**

Map:

Input:

Key: LineId

Value: head\_word<TAB>syntactic-ngram<TAB>total\_count<TAB>counts\_by\_year

Output:

Key: pattern

Value: NounPair

Reduce:

Setup: Getting from the Configuration the DPMin variable.

Input: The output of the previous map.

Output:

Key: NounPair

Value: index (a static int of the ReducerClass, represnts a unique pattern).

Cleanup: Wring to context (as a NounPair) the number of last index (= vectorSize).

**Step2:**

AnnotatedMap:

Input: hypernym.txt

Output:

Key: The NounPair parsed from the file.

Value: "-100" (will not be used)

Map:

Input: The output of the previous Reduce.

Output:

Key: NounPair

Value: count,index(of the pattern)

\* If it is the line that indicates the vectorSize, we write it to the context without changes.

Comparison:

We need to sort the lines by the following order (describing the value of the keys):

1. key = vectorSize

2. lexical order of w1 of the key

3. lexical order of w2 of the key

4. if the count of the key == -1

Reduce:

Input: 3 types of input

1. If the key equals "vectorSize", we set the total num of features to be its value. (this will be the first line we receive).

2. A NounPair that came from the AnnotatedSet (recognized by count == -1).

3. A NounPair that came from the reduce of Step1.

Output:

Key: A NounPair.

Value: A list of the features (represented by a number for each index of possible feature), seperated by a comma.

**▪ Communication**

The number of key-value pairs, and their size, sent from the mappers to the reducers-  
Step1 – **14020372** KV pairs were sent from the mappers to the reducers, with a total size of **149507589** bytes.  
Step2- **14124416** KV pairs were sent from the mappers to the reducers, with a total size of **128019314** bytes.

**▪ Results**

Precision: 0.906509253350351

Recall: 0.9796551724137931

F1 measures: 0.9416639045409347

**▪ Analysis**

True-positive:

1. abscess - symptom

2. Alabama – river

3. adulthood - period

4. affair - relationship

5. a - letter

False-positive:

1. cat - climb

2. crib - timber

3. concept - writer

4. action - roof

5. birth - larva

True-negative:

1. africa - peninsula

2. aircrew - command

3. andrew - respect

4. air - humour

5. action - dominican

False-negative:

1. blue - color

2. chicken - fowl

3. corn - grain

4. ballad - music

5. corner - place

It is obvious why it classified the **TP** pairs, as they clearly have a hypernym relation between them.  
Similarly the **TN** pairs don’t have any hypernym relations, although some of them are words that are likely to be said in the same context ( aircrew-command for example).  
As for the **FP** pairs, some of them are closely related words which probably made the classifier think they have an hypernym relation (cat – climb and crib – timber), but some of them are harder to explain due to lack of context(action – roof).  
For some of the **FN** pairs it seems that their hypernym relation is ambiguous (ballad-music for example, because ballad is a type of song, which is considered as music). Other reason could be uncommon words (chicken – fowl). As in any other category of pairs, the reason for some of the false negatives is hard to understand (blue – color which is an obvious example of hypernym relation).