ECS640U – Big Data Processing
Coursework 1 – Twitter Analysis

Dolica Akello-Egwel / 120686298

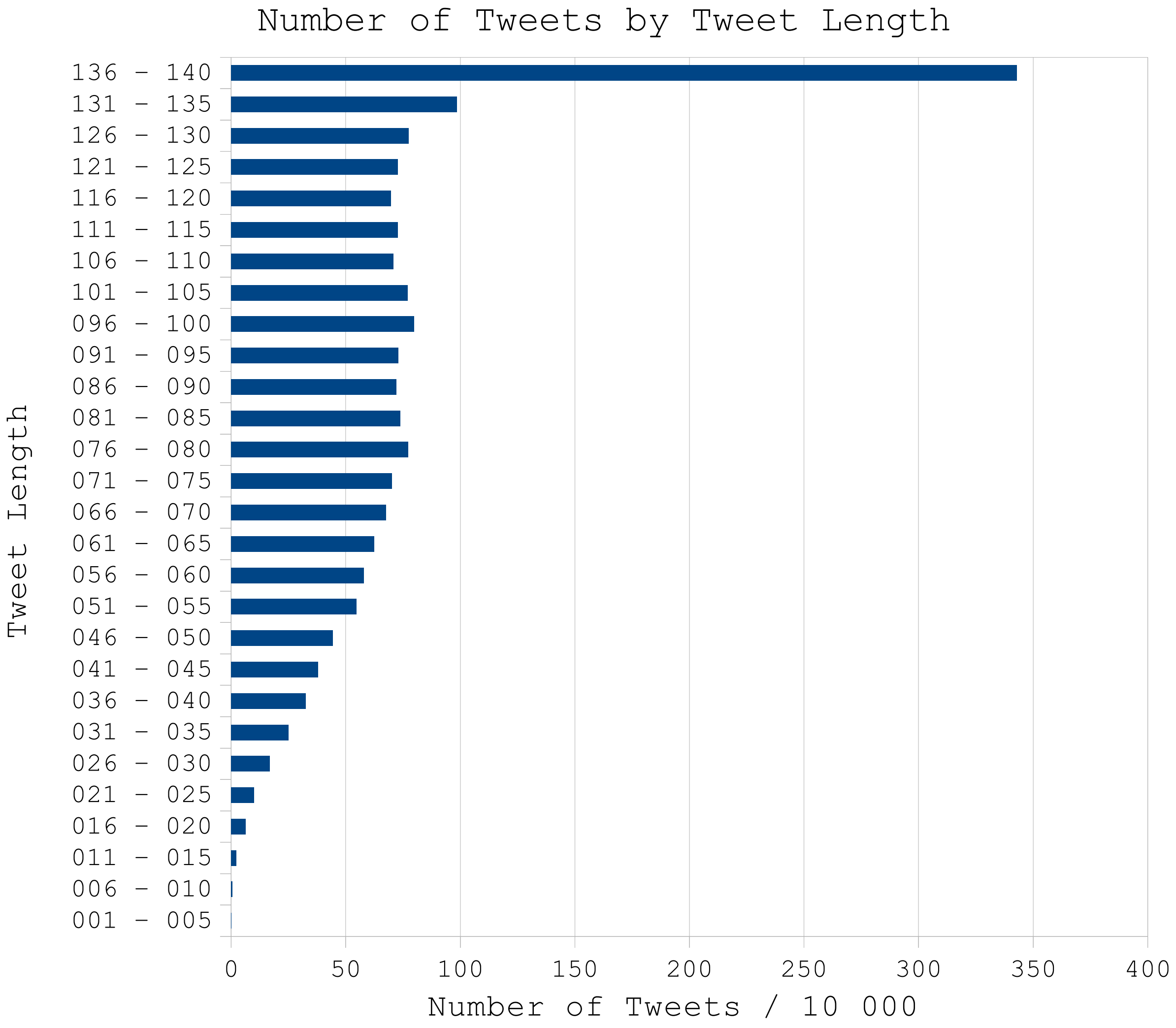
A. Content Analysis

A1. Distribution of Tweet Lengths

Program Description:

In order to arrive at the data below I used a Map program that determined the length of a tweet and then rounded this figure up to the closest five. This figure was then used to create a Key in the form of a Text object containing a String in the format [(X-4) – (X)] where X represents the closest five. The Value that the Mapper used was a single IntWritable object containing the number one. The Reducer then iterated over like Keys and used this to arrive at how many tweets in total were of length 1-5, 6-10, etc, and stored this figure in an int variable. The same Key used by the Mapper was also used by the Reducer. The final Value of the Reducer was another IntWritable object containing the sum of the various tweets of a particular length range.

Results:



Anomalies:

Tweet Length:	Number of Tweets:
141 - 145	996085
146 - 150	1268
151 - 155	73
156 - 160	57
161 - 165	57
166 - 170	21
171 - 175	17
176 - 180	21
181 - 185	31
186 - 190	16
191 - 195	38
196 - 200	17
201 - 205	16
206 - 210	56
211 - 215	8
216 - 220	4
221 - 225	9
226 - 230	8
231 - 235	4
236 - 240	2
241 - 245	6
246 - 250	10
251 - 255	5
256 - 260	4
261 - 265	6
266 - 270	4
271 - 275	3
276 - 280	2
281 - 285	1
286 - 290	2
291 - 295	3
301 - 305	2
306 - 310	2
321 - 325	1
331 - 335	1
336 - 340	1
341 - 345	1
346 - 350	1
356 - 360	1
376 - 380	1
426 - 430	1
481 - 485	1
541 - 545	1
546 - 550	2
551 - 555	1

In total there were 997 871 tweets with a length greater than 140 characters. The majority of these tweets were concentrated in the 140 – 145 range (996 085). I chose to omit this from the graph because it is possible that some of these tweets were in truth only ~20 characters long but were interpreted incorrectly due to containing non-alphabetic characters. Due to it being difficult to determine what exactly caused these abnormal values and to accurately guess their true length it seems best to simply present this information in a separate table rather than skewer the other results. It is also possible that some

A2. Average Tweet Length

<u>Program Description:</u>

The Mapper program processed the Twitter data by creating a Key which contained a String with the value "Average:" for every tweet. The Key was an IntWritable object stored an int of the tweet length. The Reducer then iterated over the whole collection of tweets and added their lengths then divided this by the size of the collection. For this purpose the "sum" and "size" values were cast as doubles and the result was stored in a DoubleWritable object, which was the Key output of the Reducer.

<u>Results:</u>

Average: 98 (Rounded to the nearest character)

<u>Anomalies:</u>

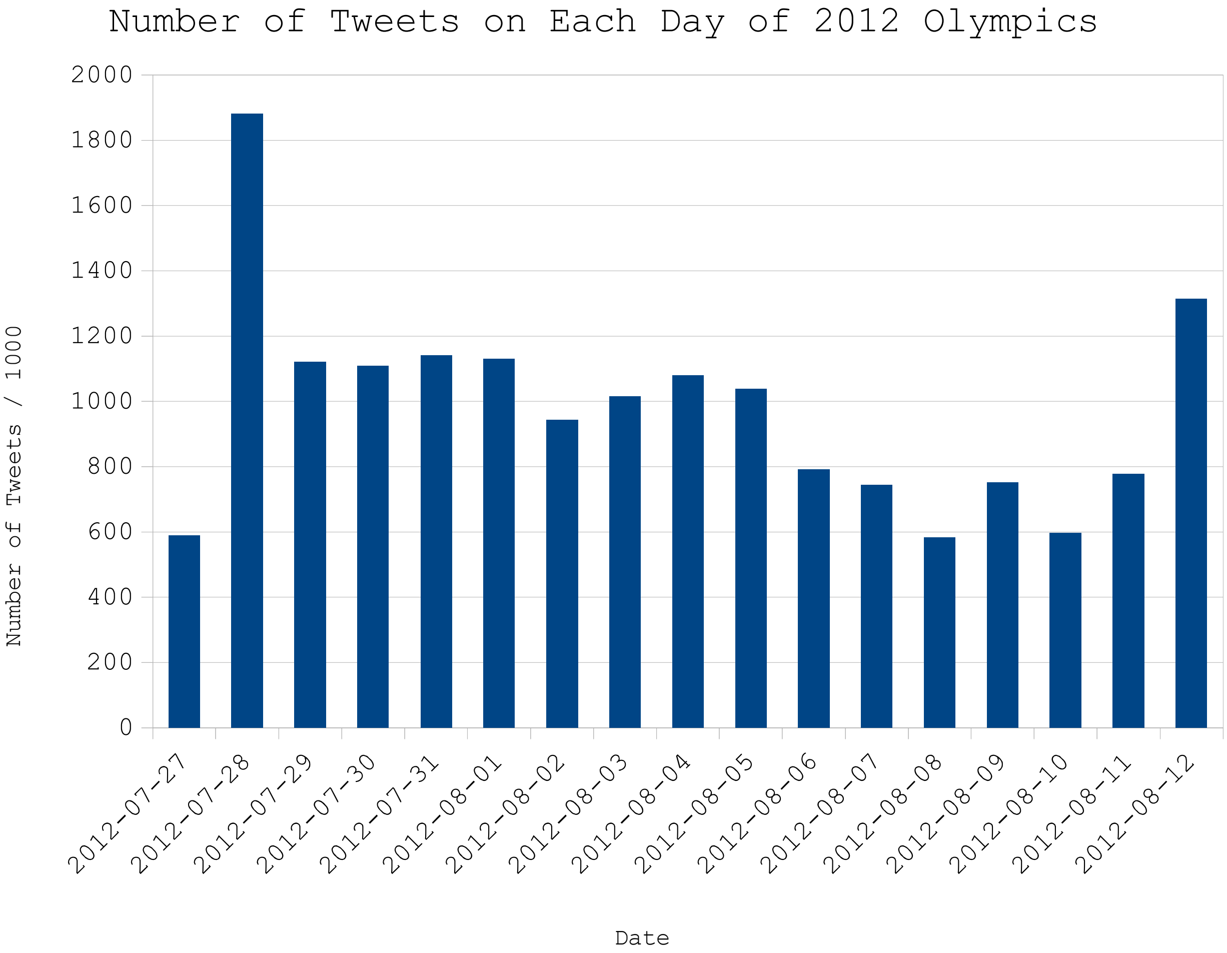
Tweets with a length greater than 140 were discarded by the Mapper so as not to distort the results. This was achieved by having the Mapper only call the write() method on the context if the length of the tweet was smaller than or equal to 140 characters.

B. Time Analysis

<u>Program Description:</u>

The Mapper method isolated the date within the tweet data by trimming the String from the first semi-colon character to the point in the String 10 characters after the first semi-colon. This was then used as the Key in the form of a Text object. The value was a IntWritable object containing the number one to reflect a single occurrence of a tweet from this particular date. The Reducer then iterated over this information to find the total number of tweets for each day.

Results:



C. Hashtag Analysis

Program Description:

The Mapper program works by taking a String of the hashtag segment of the tweet. It is then converted to lower case and its non-alphabetic characters are removed. Two arrays are then used to determine if any hashtags contain support for a particular country. The first array contains Strings of support phrases in English, Spanish, Portuguese, German, French, and Polish such as “go” and “viva”. The second array is two-dimensional: each row being an array containing a String of a country name, its 2 and 3-character country codes, its names in different languages, etc. The English-language name of the country always appears first in the row. In the cases of names in different languages where non-alphabetic characters were present, these were removed, i.e. “espaa” is used rather than españa. This is so it is consistent with the output produced when the hashtag String is converted to lower case and all characters besides spaces and a-z are removed.

With these arrays a series of loops are then used to determine if the hashtag segment contains some combination of a supportive phrase and a country name. If a match is found then a Text object containing the country's English-language name used as the Key. The Value is an IntWritable object storing the number one.

Finally, the Reducer iterates over these to find out the total number of supportive tweets there were for each country. The Reducer output is the Text-object Key of the country's name, and the IntWritable of the total number of supportive hashtags.

Program Limitations:

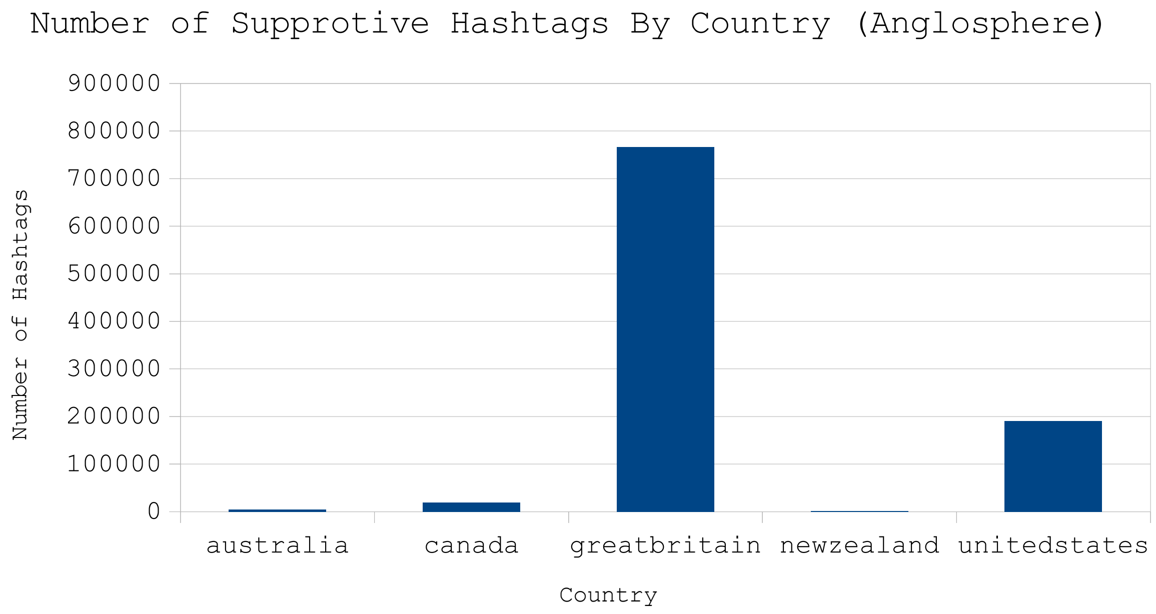
Chinese is the most widely-spoken language in the world but was not accounted for by the program. Many other languages were also left out of the process. However, because Spanish, English, and French are so widely spoken in the world it was hoped that this would still present a lucid picture of which teams were being rooted for the most. Another issue is that it was difficult to determine what exactly supportive phrases were in various languages.

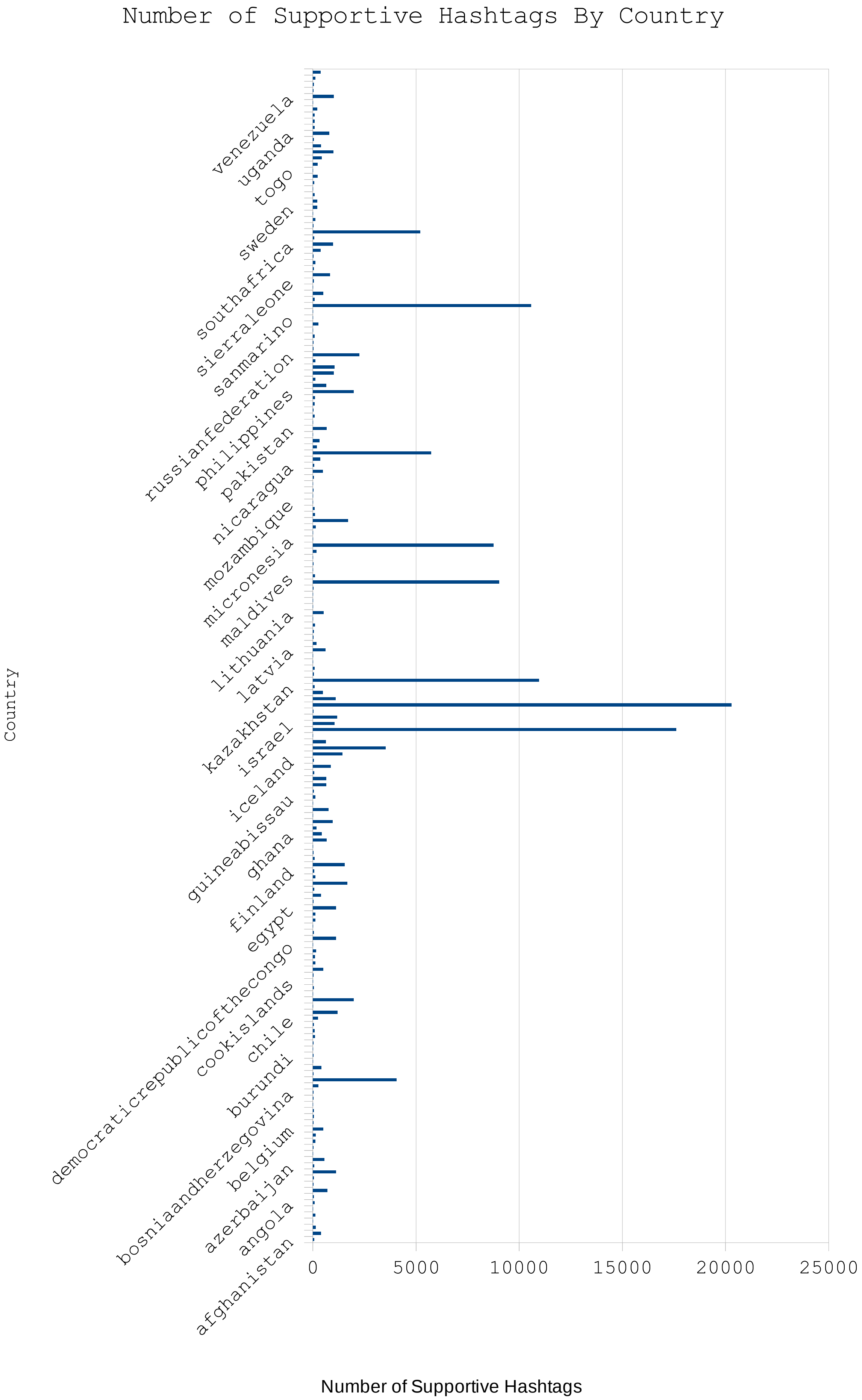
To avoid mistaking “gousainbolt” as support for America it was necessary to check if “ teamusa ” (with spaces on either side) rather than “teamusa” was a substring of the hashtag segment. This approach also meant that any hashtags that happened to be in the format of “teambritainforever” would not be viewed as support for Britain. It was hoped that these accounted for a minority of the supportive tweets, and that missing them out would still provide a helpful understanding of which teams people were supporting.

This approach also meant that someone might feature the same hashtag twice and it would be viewed as one instance of support. But differing combinations of supportive phrases would be viewed as multiple instances of support if contained in the same tweet.

Because the results were biased towards English, it was necessary to separate English-speaking countries and non-English speaking countries.

Results:


Number of Supprotive Hashtags By Country (Anglosphere)
Number of Hashtags
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
australia
canada
greatbritain
newzealand
unitedstates
Country


Number of Supportive Hashtags By Country
Country
venezuela
uganda
togo
sweden
southafrica
sierraleone
sanmarino
russianfederation
philippines
pakistan
nicaragua
mozambique
micronesia
maldives
lithuania
latvia
kazakhstan
israel
iceland
guineabissau
ghana
finland
egypt
democraticrepublicofthecongo
cookislands
chile
burundi
bosniaandherzegovina
belgium
azerbaijan
angola
afghanistan
0
5000
10000
15000
20000
25000
Number of Supportive Hashtags