Detecting Political Bias In News Channels

Under Guidance Of:

Prof. Nirmal Kumar Sivaraman

Submitted By:

Jigyasa Yadav 16ucs084
Sakshi Srivastava 16ucs164
Ujjwal Madan 16ucs204

Table of Contents

Introduction

- The role of news agencies is to provide factual, neutral and unbiased news without any opinion of the agency or the person reporting it. This is what helps the government and the society at large to be aware of causes, reactions and effect. Clean and clear media is a major contributing factor to help a nation progress further.
- We consider bias in news reporting as presenting the news in such a way that force people to think about the causes, effects and the reactions to the events in a certain way.

Media bias is classified into three types

Visibility bias,refers to the political actors being mentioned in the news

Tonality bias, refers to the evaluation of these actors

Agenda bias, refers to the extent to which news addresses issues that the parties prefer.

Existing Work

- **Collection of tweets using R**
- **Selection of top 3 news channels based on their retweet counts only and extraction of top 10 hashtags**
- **Perform mapping of news channel manually.**
- **Calculating Similarity score for each hashtag with each political party.**
- **Calculating sentiment score of each news channel for each hashtag**
- **Then calculating the bias score for each news channel corresponding to each political party.**

Problem Statement

To automate the process of mapping a tweet with the most relevant hashtag among all the hashtags the tweet contains using Python

Drawbacks of previous work

- ❏ Manual mapping of tweets to hashtags can be a very time expensive (consuming) task.
- ❏ Tweet collection was to be run manually every time.
- ❏ News channels selected by them has retweet count as the only decision parameter which does not gives an accurate measure of popularity.

Selecting news channel

- Select popular indian english news channels .

Data Collection

Collecting tweets from all the popular news channels.

(automated)

Data Exploration

Explore several features about each news channel

Such as retweet count, avg tweet count per day etc.

Selecting top 3 news channels

- Selecting top 3 news channels based on score of news channel.
- 2.5(retweet_count) + (likes_count) will b the score for each news channel.

News Channel	Average Retweets	Average Likes	Average Tweets Per Day
Republic	30	78	268
ABP	31	128	110
Zee News	26	130	72
Times Now	18	47	349
NDTV	11	44	341
Times Of India	11	55	249
India Today	15	49	265
CNBC	19	31	218
DD News	403	115	57
CNN	9	15	299

2.5(retweet_count) + (likes_count) (score for each news channel)

DD News

We can clearly see that 75% of the data has Retweet count less than 54 But because of some outliers the average retweet count is high.

In [56]:
```
DD.describe()
```

Out[56]:

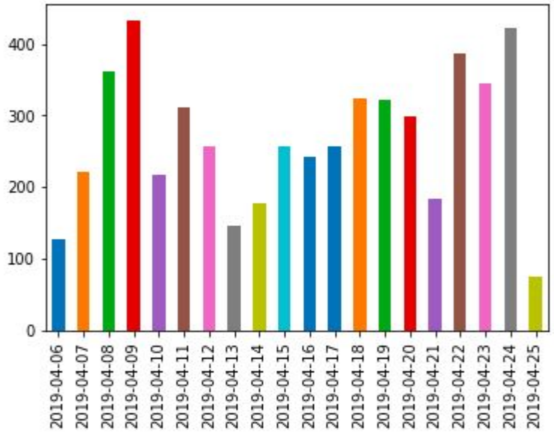
	id	retweet_count	favorite_count
count	5.045000e+03	5045.000000	5045.000000
mean	1.103970e+18	403.316353	115.725867
std	1.060814e+16	2150.349957	281.027467
min	1.085495e+18	0.000000	0.000000
25%	1.094537e+18	8.000000	19.000000
50%	1.103691e+18	19.000000	50.000000
75%	1.113542e+18	54.000000	119.000000
max	1.121342e+18	66533.000000	10408.000000

In [57]:
```
print("Avg tweets per day:",df.mean())
```

```
Avg tweets per day: 57.9
```

REPUBLIC News

```
In [6]: df=republic['created_at'].value_counts()
df=df.sort_index()
df=df.tail(20)
df.plot(kind='bar',x='created_at',y ='tweet count')
plt.show()
```



No. of tweets posted for past 20 days

```
republic.describe()
```

	id	retweet_count	favorite_count
count	1.457600e+04	14576.000000	14576.000000
mean	1.111954e+18	29.657108	78.020582
std	5.531497e+15	114.297861	219.226612
min	1.101479e+18	0.000000	0.000000
25%	1.107509e+18	4.000000	19.000000
50%	1.111910e+18	9.000000	36.000000
75%	1.116607e+18	23.000000	75.000000
max	1.121350e+18	4776.000000	9941.000000

```
print("Avg tweets per day:",df.mean())
```

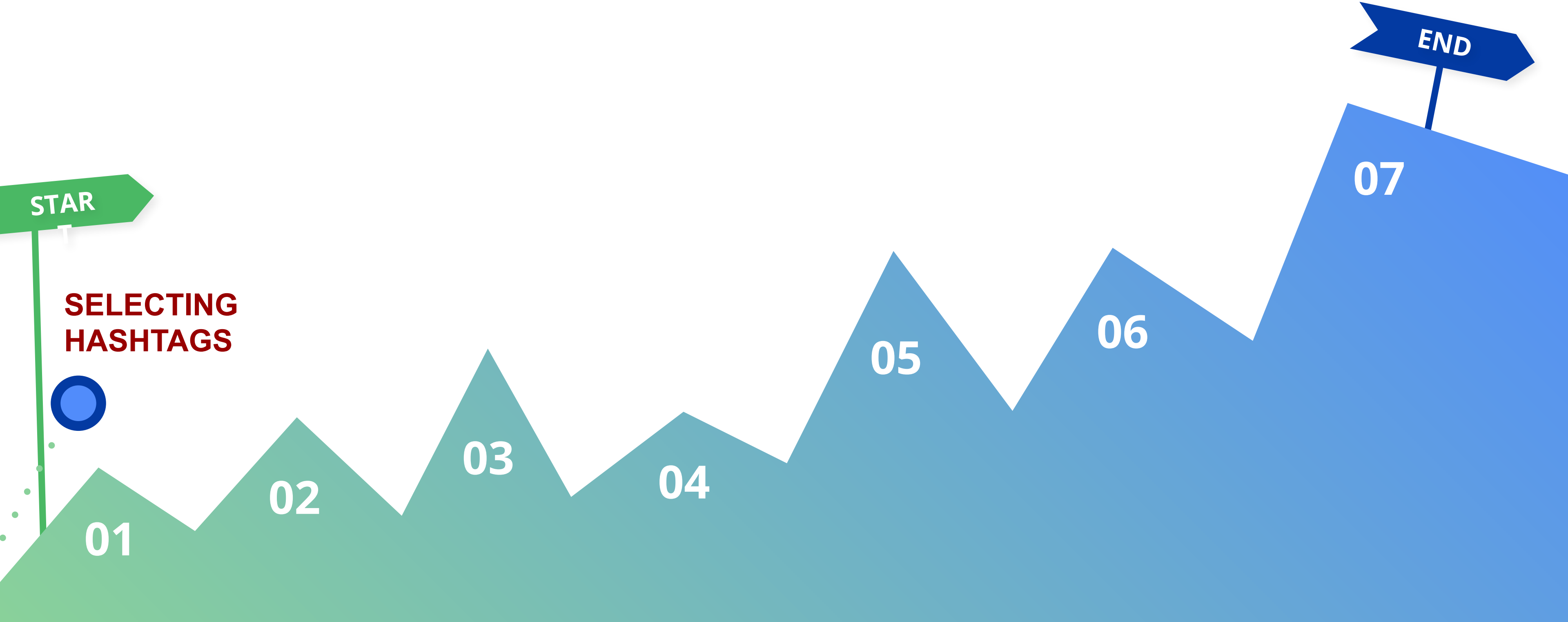
```
Avg tweets per day: 268.35
```

Simulation Snippet

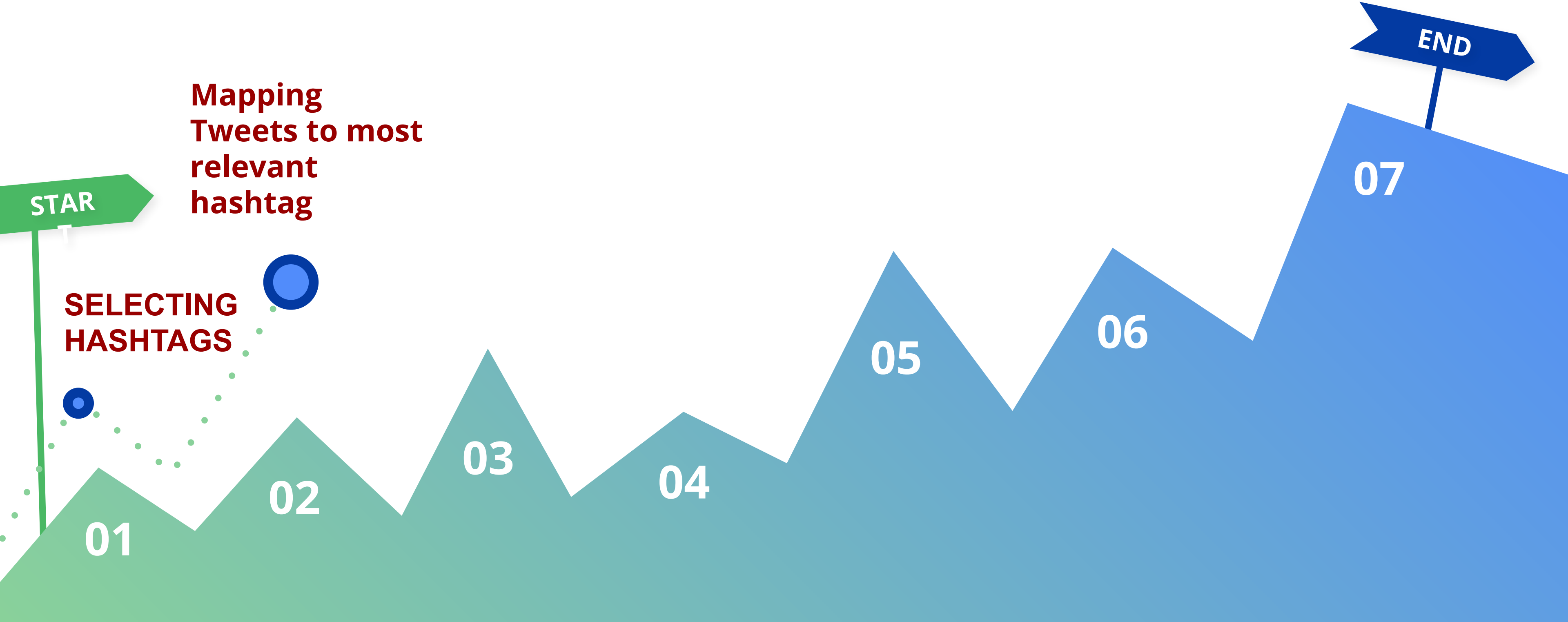
Total Tweets collected are 2,40,000 from 10 English Indian news channel's Twitter handles:-

- Republic
- ABP
- Zee News
- Times Now
- NDTV
- Times Of India
- India Today
- CNBC
- DD News
- CNN

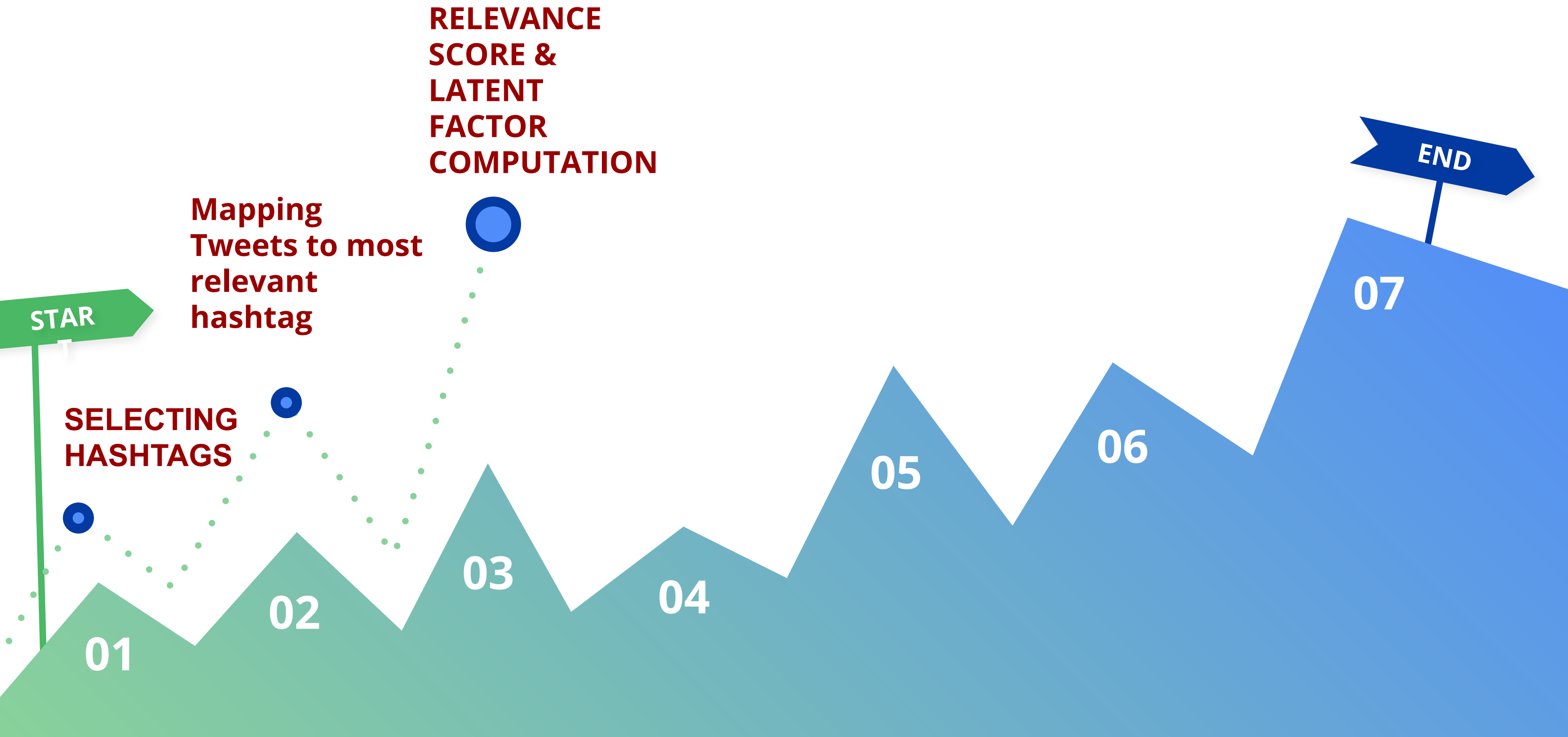
7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



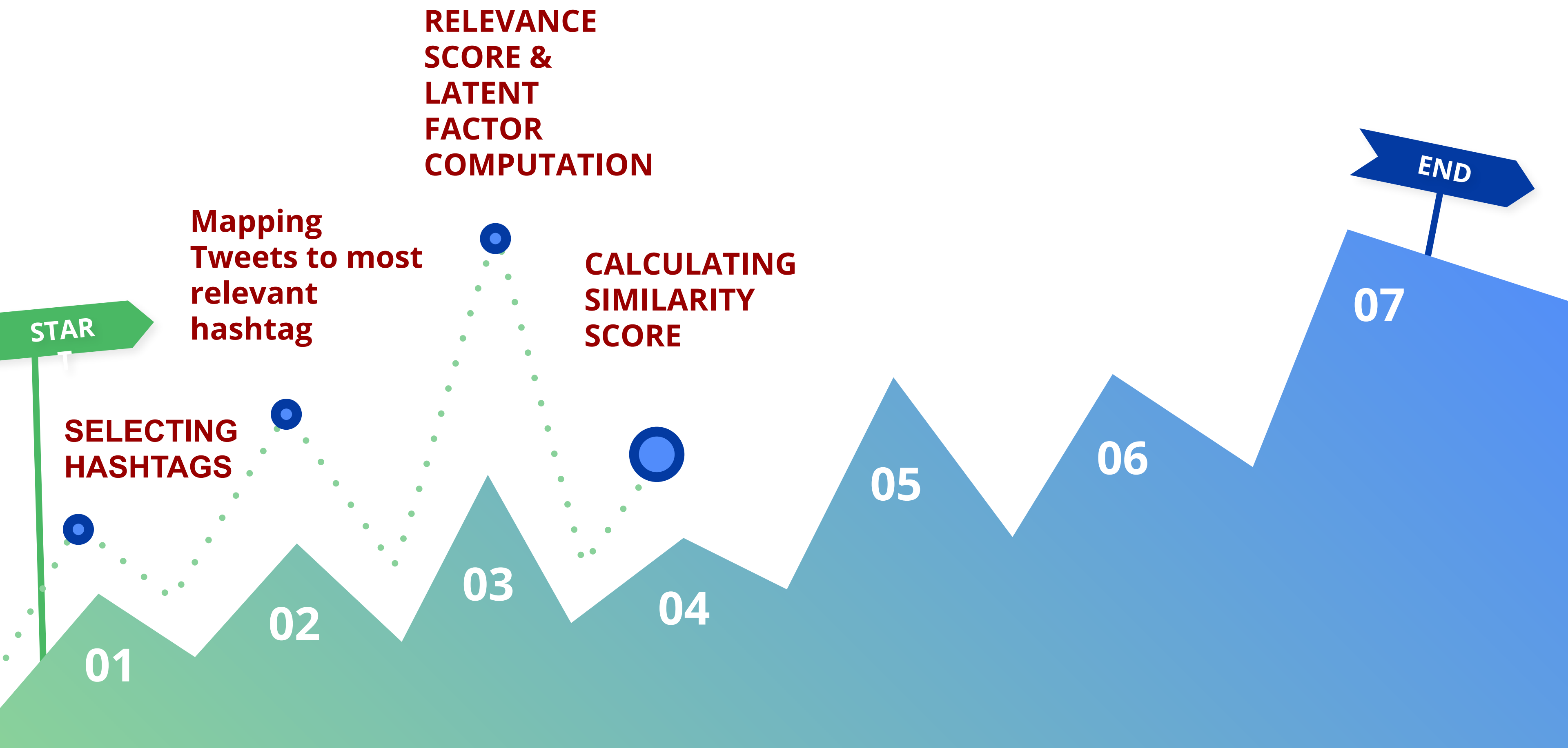
7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



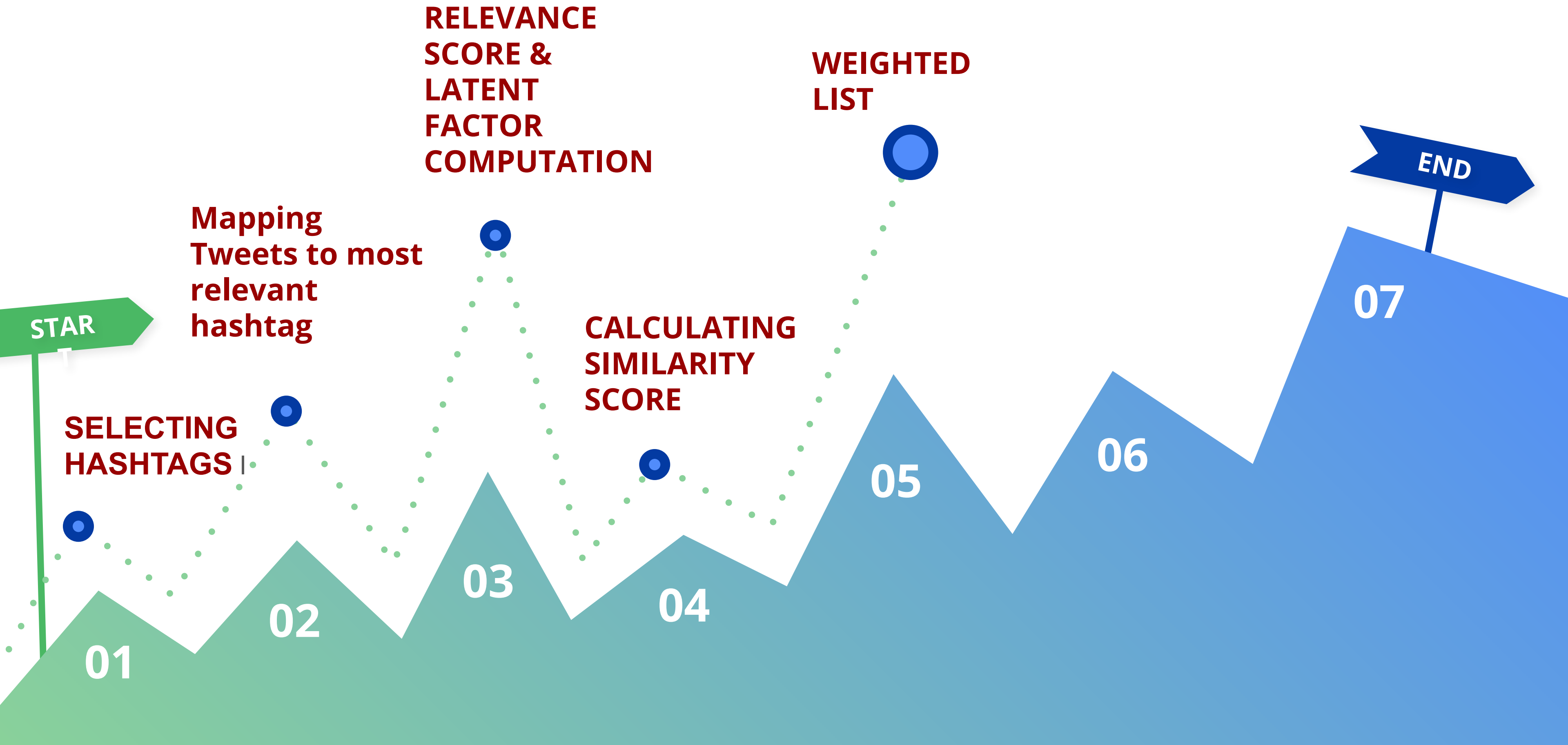
7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



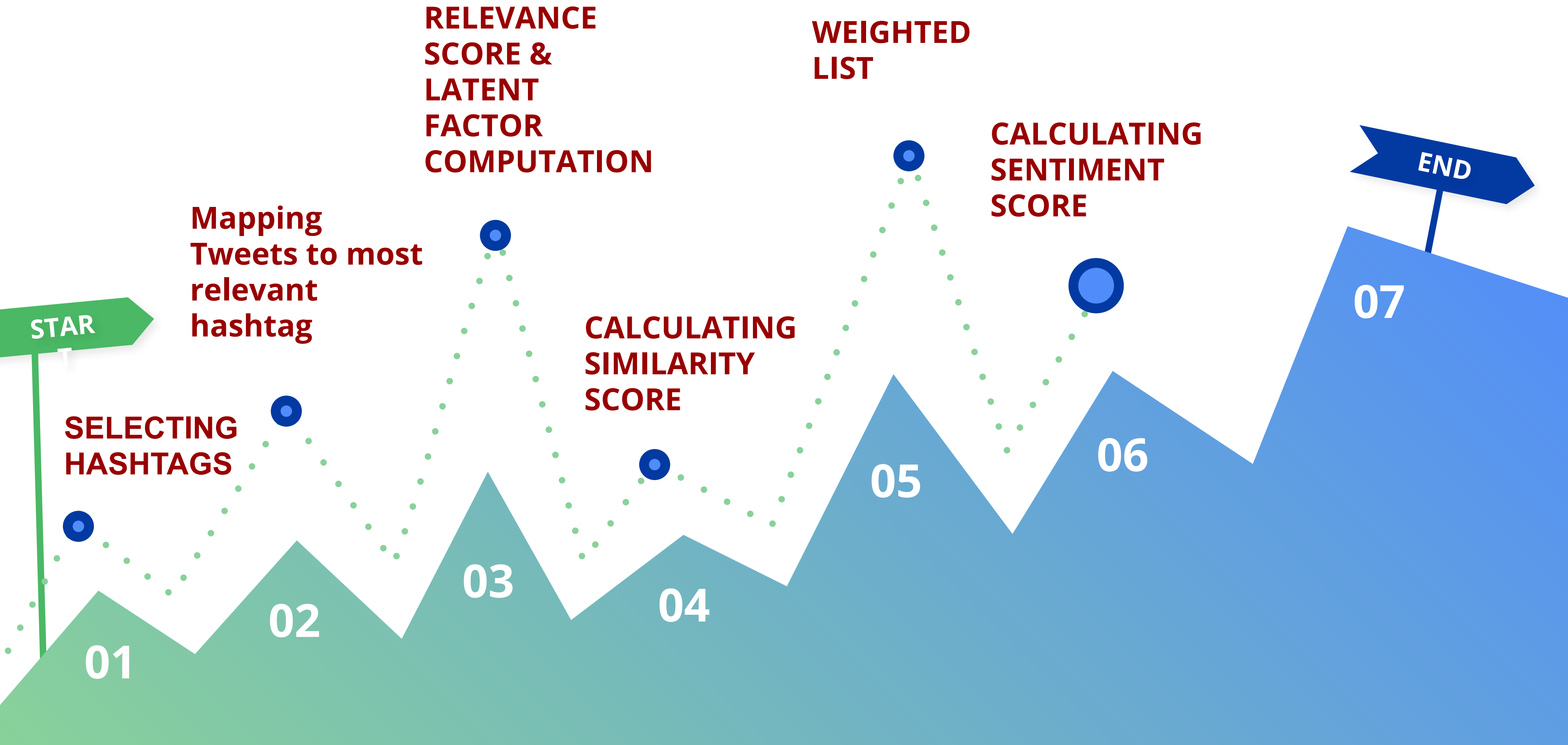
7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



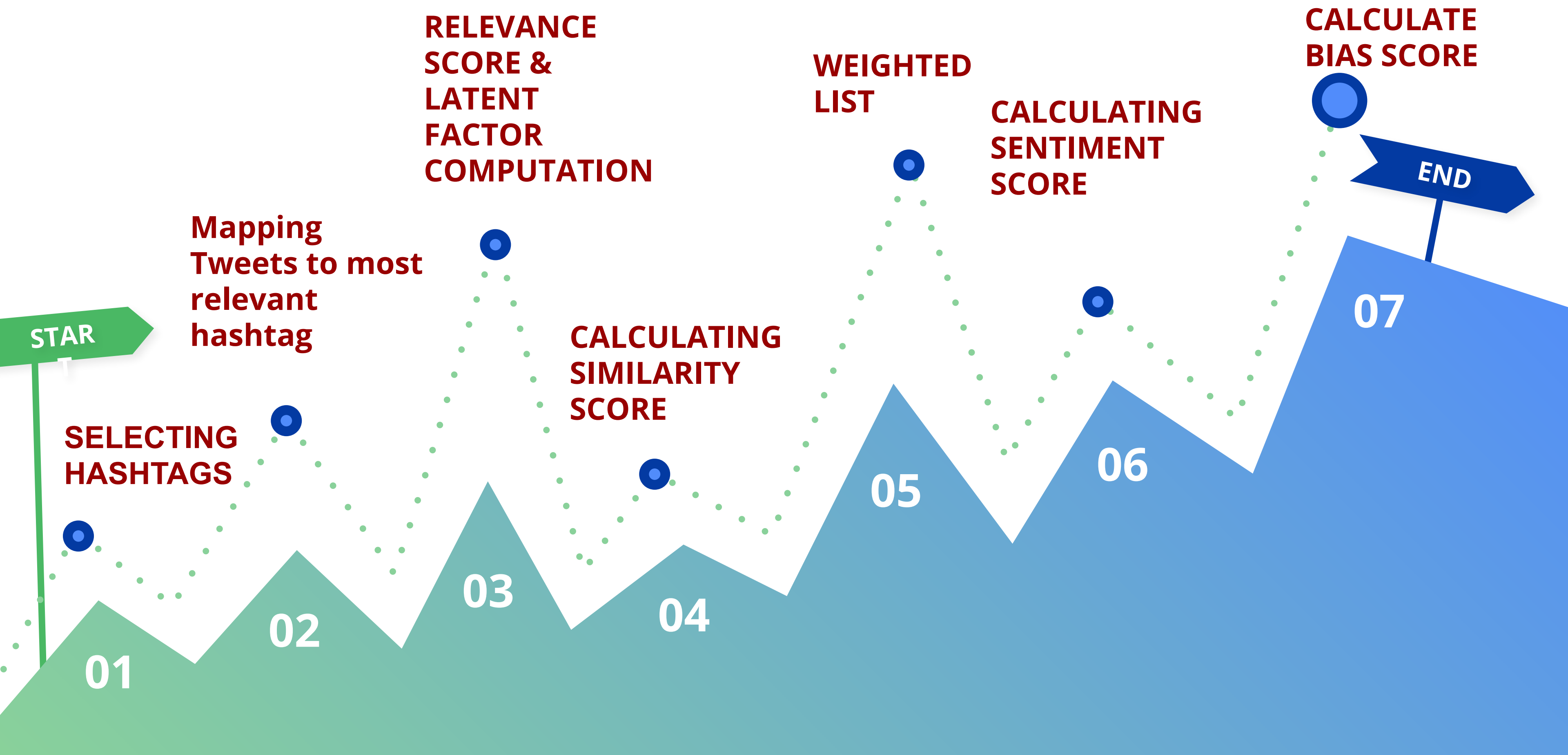
7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



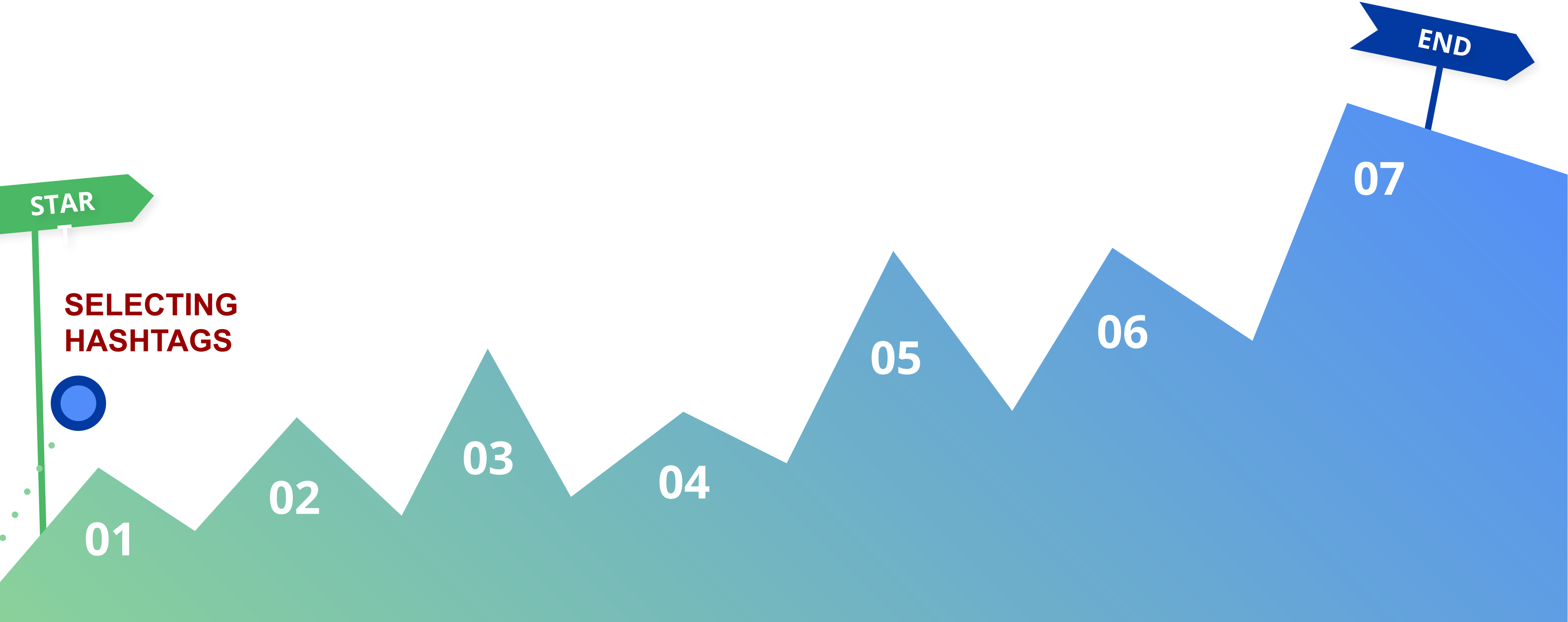
7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



7 Step Mountain Journey to map each tweet with a hashtag
And finding the bias

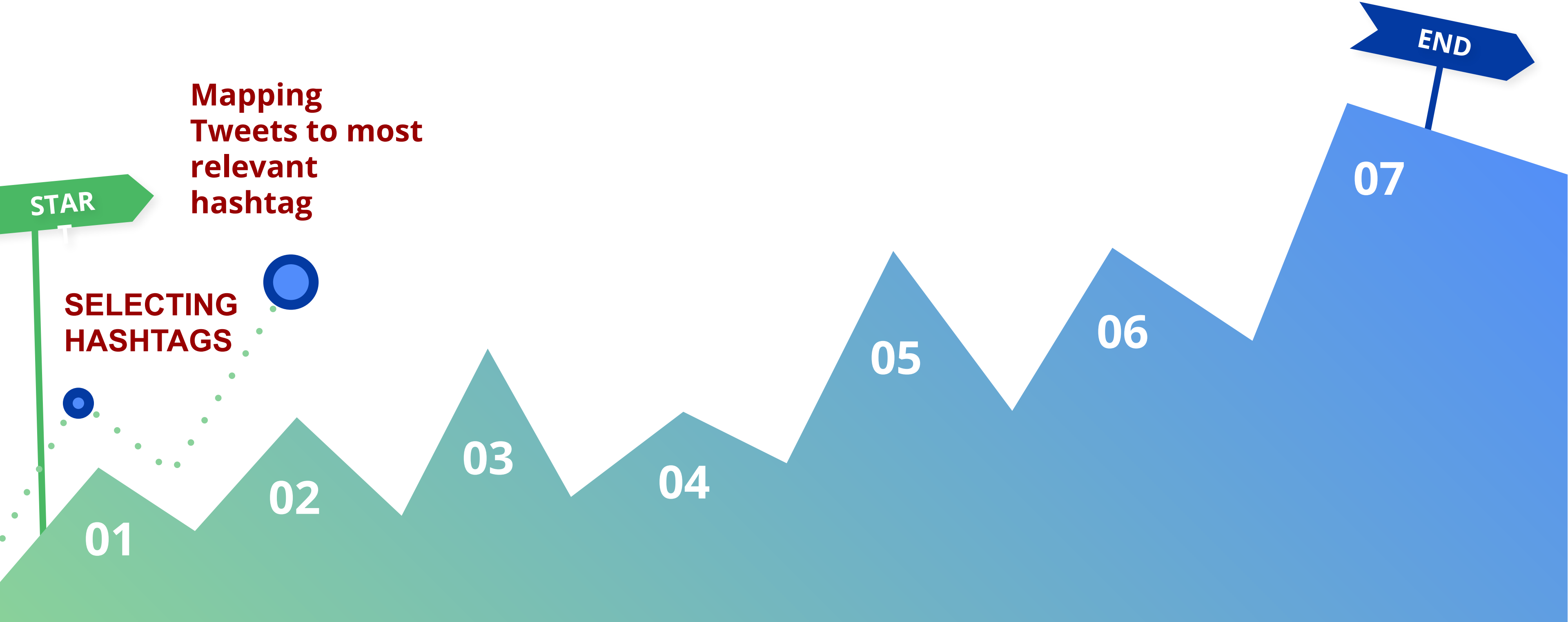


Selecting Top ten Hashtags

- Finding frequency of tweets for each hashtag.
- Automated mapping of related hashtags to one common
  general hashtag.
- Finding common hashtags among news channels.
- Selecting top 10 hashtags from the list of common hashtags

#Abhinandan
#modi
#LokSabhaElections
#Worldcup
#Budget
#Congress
#bjp
#antihindipolitics
#mumbairain
#kashmir

7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



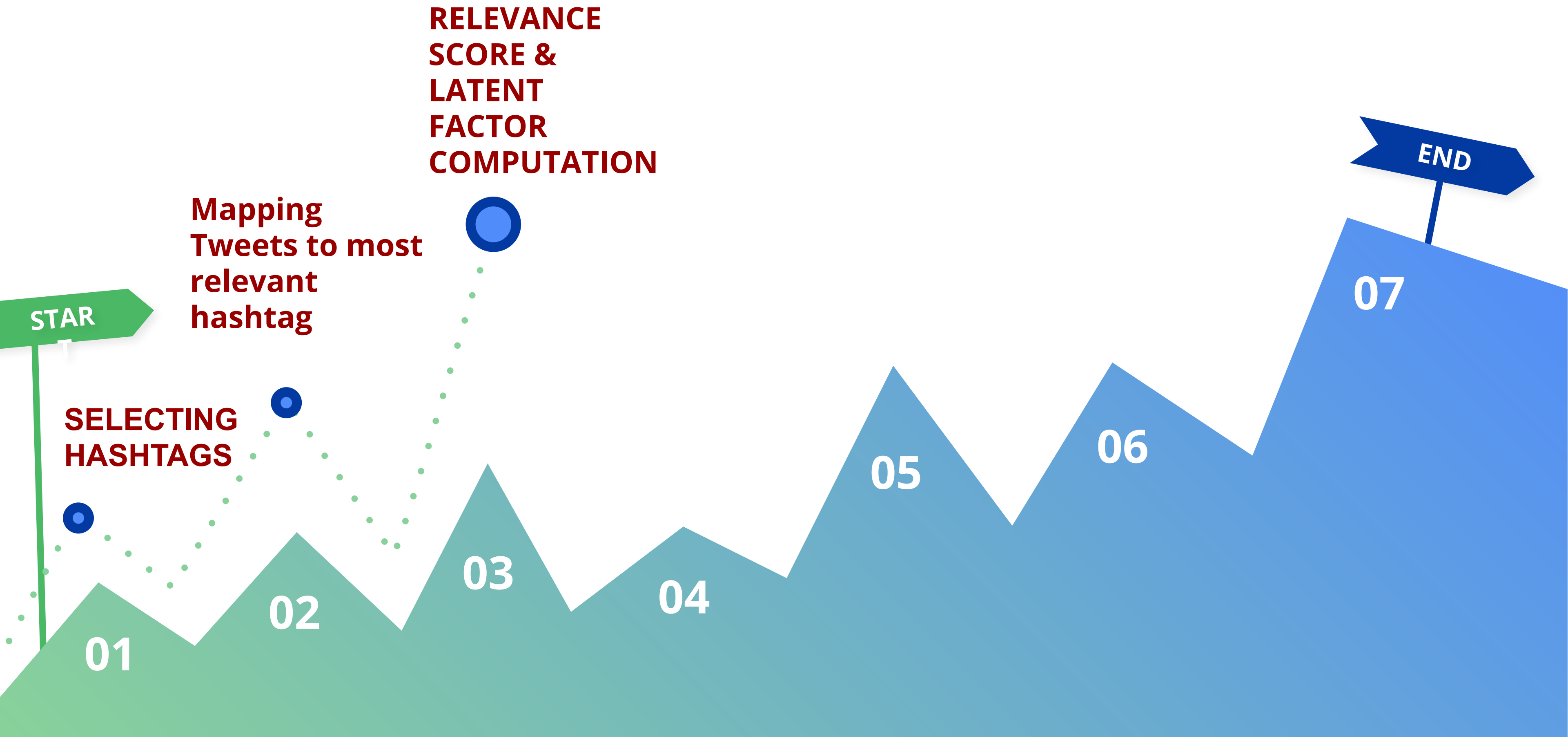
Previous Mapping

Previously the mapping was done manually, each tweet was mapped to whichever hashtag,the tweet contains, seemed to be more relevant to it.

Automated Mapping

The mapping this time is also done based on relevancy,but this time relevance score is computed and whichever hashtag has the highest relevance score,the tweet will be mapped to that particular hashtag only.

7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



Relevance Score Computation

Relevance score between a hashtag and a tweet can be given by the formula:

$$Rel(h,d) = [\sum_{i=1}^{k^{(w)}} \alpha_i^{(w)} \mathbf{w}_i^T + \sum_{i=1}^{k^{(l)}} \alpha_i^{(l)} \mathbf{l}_i^T + \sum_{i=1}^{k^{(m)}} \alpha_i^{(m)} \mathbf{m}_i^T]\mathbf{h}$$

where,

- wi , li , mi , h represent the latent factors for word wi , link li , mention mi
- α (w) i , α (l) i , and α (m) i are weights of each latent vectors.

What is TF-IDF?

The most widely used techniques to process textual data is TF-IDF.**TF-IDF** stands for “Term Frequency—Inverse Data Frequency”.

Term Frequency (tf): gives us the frequency of the word in each document .It is the ratio of number of times the word appears in a document compared to the total number of words in that document.[7]

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

INVERSE DATA FREQUENCY (IDF):

used to calculate the weight of rare words across all documents.[7]

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Combining these two we come up with the TF-IDF score (w) for a word.

Choice of α(*)

- For terms, αi for an ith word wi is defined to be TF-IDF(wi)
- Since most tweets contain one or two links or mentions, α (l) i and α (m) i are defined to be the reciprocal of k (l) and k (m) , respectively. In other words, links and mentions are both equally weighted.

Latent Factor Computation

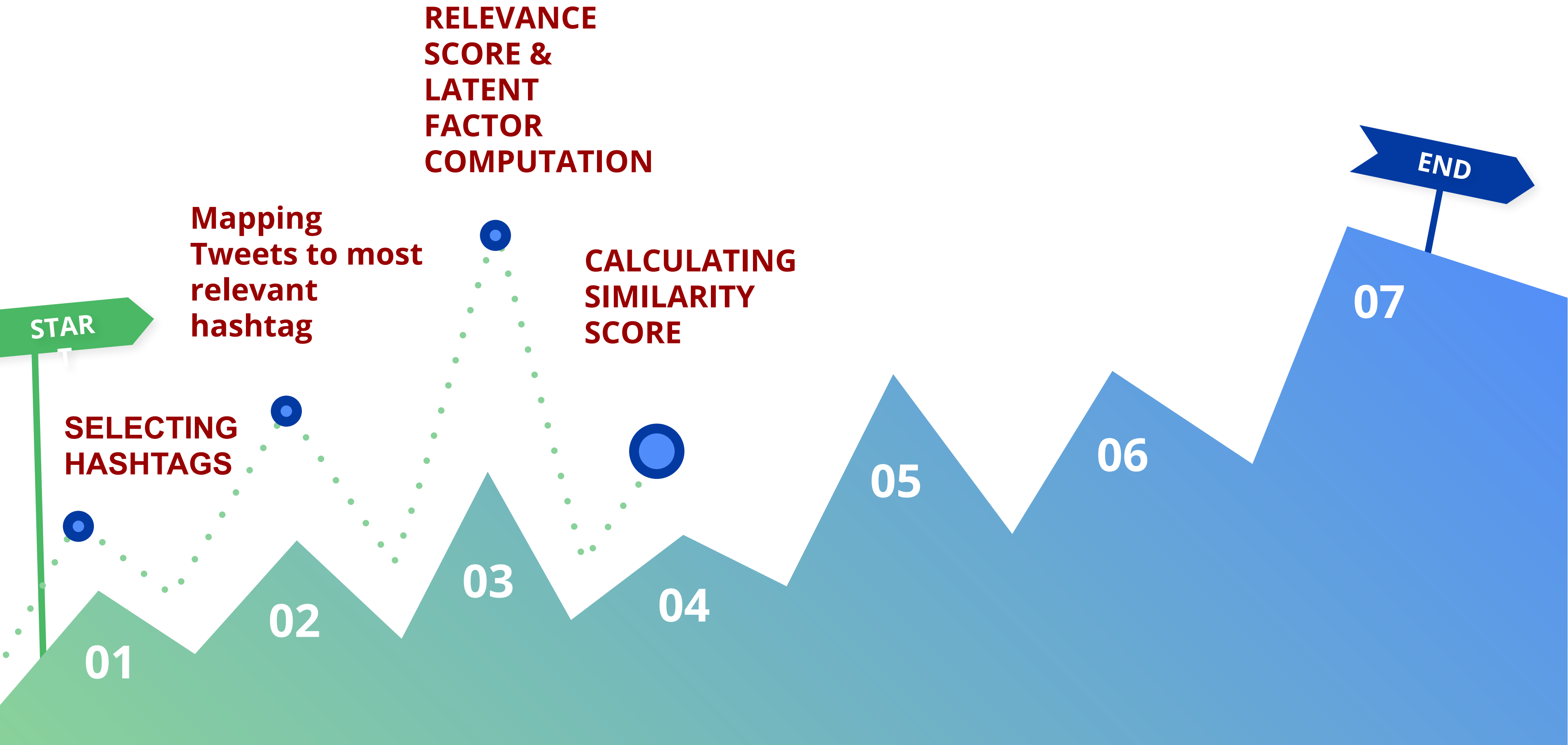
	word1	word2	word3	word4
Feature1	x			
Feature2	y			

	Feature1	Feature2
Hashtag1	a	b
Hashtag2		
Hashtag3		

	word1	word2
Hashtag1	value1	
Hashtag2		
Hashtag3		

value1 = x*a+y*b

7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



Similarity Score between political party and a hashtag

❖ For every hashtag h, give a similarity score with each political party p using below explained algorithm. Let this be denoted as Similarity(h, p).

Computing Similarity between political party and hashtag

- For each hashtag, generate a weighted list of terms by considering all the tweets labeled with that hashtag.The weighting scheme used is TF-IDF.
- Generate a weighted list of terms for a political party considering term extracted from sources such as official websites of the party and wikipedia pages of the party.
- A similarity score is found between the two lists using Pearson correlation coefficient.

$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}}$$

Where:

- N = number of pairs of scores
- Σxy = sum of the products of paired scores
- Σx = sum of x scores
- Σy = sum of y scores
- Σx^2 = sum of squared x scores
- Σy^2 = sum of squared y scores

Pearson Correlation Coefficient Formula

For each tweet t, find the sentiment score Sentiment(t)

Let us denote the set of tweets that are labeled with hashtag h as Th = {th1, th2, ..., thn}. For every news channel cj , find the average sentiment score for their tweets that are labeled every hashtag hi

Similarity Score

1 NDTV

```
dict_sim_ndtv_bjp={}
for i in list(dict_ndtv_weighted.keys()):
    k=numpy.corrcoef(dict_ndtv_weighted[i],bjp_weighted_ndtv)[0,1]
    dict_sim_ndtv_bjp[i]=abs(k)
```

```
dict_sim_ndtv_bjp
```

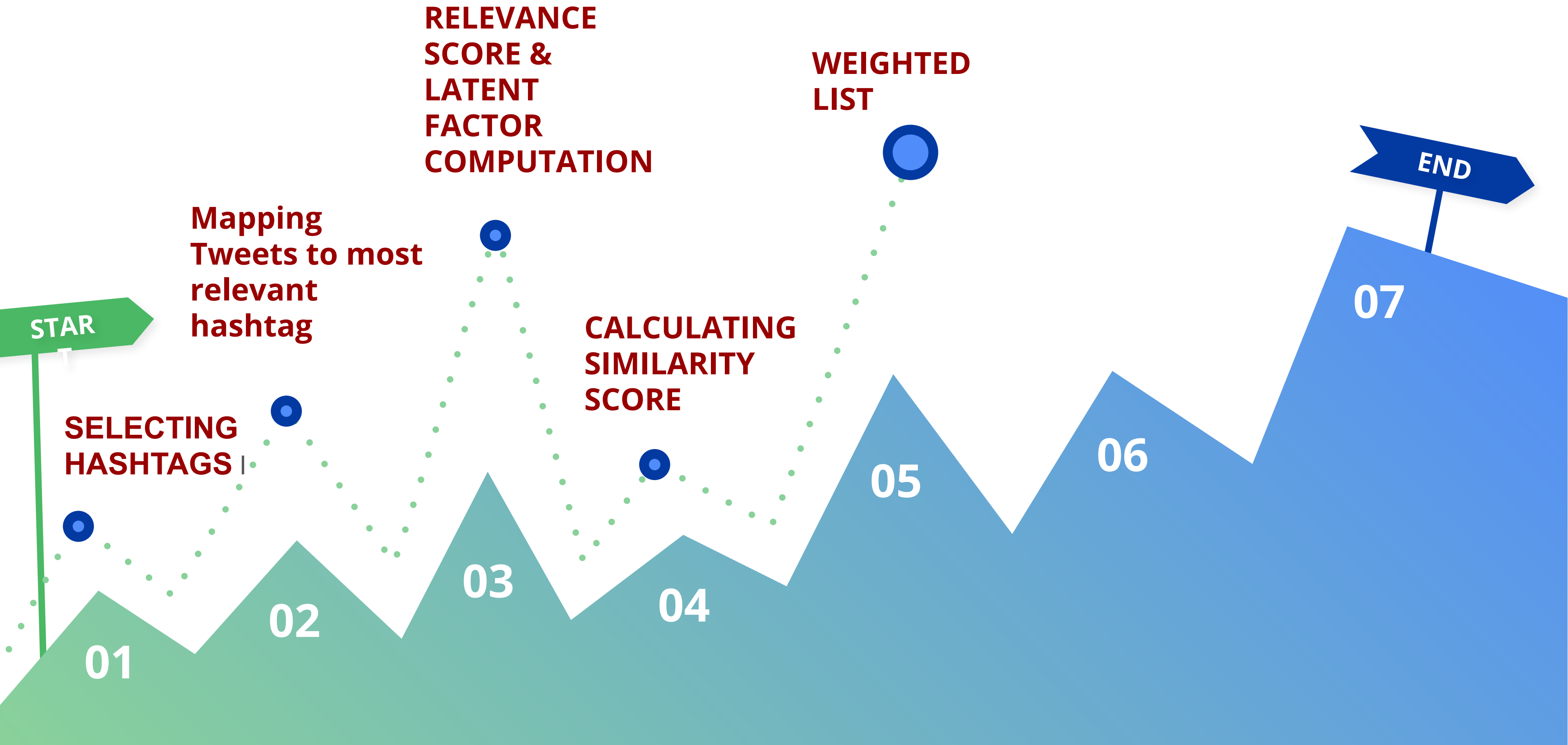
```
{'abhinandan': 0.36699978263083394,
 'modi': 0.45706839421927387,
 'elections': 0.3201717573578328,
 'worldcup': 0.027429768423657387,
 'budget2019': 0.10295609411872346,
 'congress': 0.32017175735783276,
 'bjp': 0.2937863354265258,
 'politics': 0.5187401543274971,
 'mumbairain': 0.10267247764608102,
 'kashmir': 0.45706839421927414}
```

```
dict_sim_ndtv_congress={}
for i in list(dict_ndtv_weighted.keys()):
    k=numpy.corrcoef(dict_ndtv_weighted[i],inc_weighted_ndtv)[0,1]
    dict_sim_ndtv_congress[i]=abs(k)
```

```
dict_sim_ndtv_congress
```

```
{'abhinandan': 0.08962181686777514,
 'modi': 0.39275866238067725,
 'elections': 0.5608793653983681,
 'worldcup': 0.31114687450740935,
 'budget2019': 0.3189962975678455,
 'congress': 0.2163596275967732,
 'bjp': 0.618135933818837,
 'politics': 0.4249218731977155,
 'mumbairain': 0.2583898033511959,
 'kashmir': 0.2163596275967733}
```

7 Step Mountain Journey to map each tweet with a hashtag
And finding the bias



Weighted List for Political Parties

- To generate a list of terms of a political party we scrape data from these sources and put them in the database:-
  - Official websites of the party. Contains the names of major leaders, schemes and agendas.
  - Their Wikipedia Pages.
  - Descriptions of official pages or screen names of party and party leaders on twitter itself.

Gather Important keywords from political parties ¶

```
import wikipedia
inc = wikipedia.summary("Indian National Congress")
bjp=wikipedia.summary("BJP")
```

```
list_inc=inc.split(".")
list_bjp=inc.split(".")
```

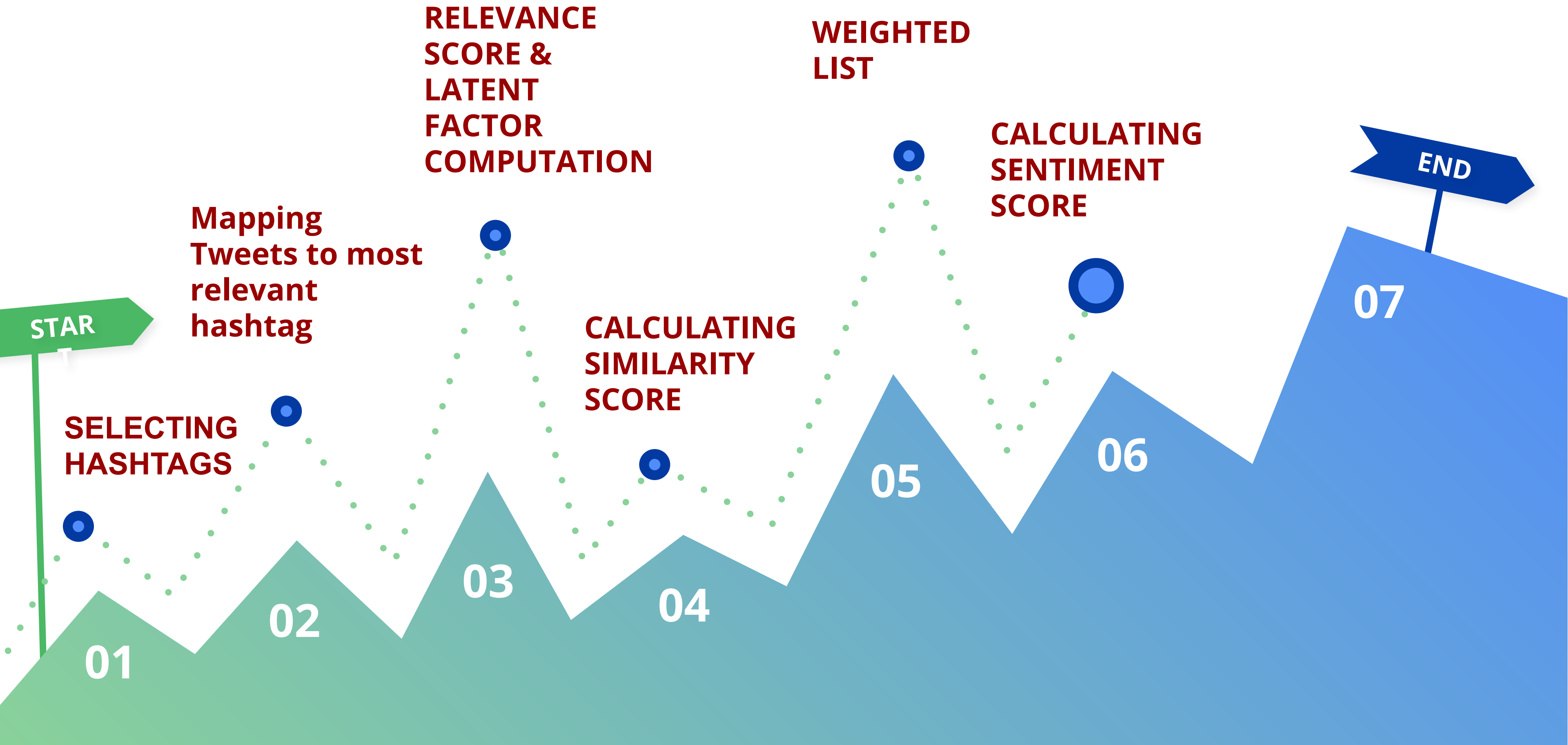
```
list_bj =['gujarat','namoagain','mainbhichowkidar','modi','indiafirst','narendra','amit','atalbihari','bjp','amitshah']
```

```
list_in=['congress','rahul','youthcongress','priyanka','gandhi','sonia','nsui','nehru','manmohan','inc']
```

```
data_inc = cv.fit_transform(list_inc)
tfidf_transformer=TfidfTransformer()
# convert term-frequency matrix into tf-idf
tfidf_matrix_inc = tfidf_transformer.fit_transform(data_inc)

# create dictionary to find a tfidf word each word
word2tfidf_inc = dict(zip(cv.get_feature_names(), tfidf_transformer.idf_))
```

```
data_bjp = cv.fit_transform(list_bj)

# convert term-frequency matrix into tf-idf
tfidf_matrix_bjp = tfidf_transformer.fit_transform(data_bjp)

# create dictionary to find a tfidf word each word
word2tfidf_bjp = dict(zip(cv.get_feature_names(), tfidf_transformer.idf_))
```

7 Step Mountain Journey to map each tweet with a hashtag And finding the bias



$$AvgSentiment(h_i, c_j) = \frac{\sum_{i=1}^{n} Sentiment(t_{hi})}{n}$$

The bias of a news channel c towards a party p for a hashtag h,will be calculated using the formula given below:

Bias(c, p, h) = AvgSentiment(h, c) × Similarity(h, p)

The final bias of a news channel c towards a party p over a set of hashtags H = h1, h2, ...hm,will be calculated using the formula given below:

Sentimental Score of Hashtags with News Channels

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

1 NDTV

```
analyser = SentimentIntensityAnalyzer()
def sentiment_analyzer_scores(sentence):
    score = analyser.polarity_scores(sentence)
    #print(type(score))
    return score['compound']

dic_ndtv_score={}

for i in dic_ndtv.keys():
    total_score=0.0
    for j in dic_ndtv[i]:
        total_score+=sentiment_analyzer_scores(j)
    l=(len(dic_ndtv[i]))
    ans=total_score
    if l !=0 :
        ans=ans/len(dic_ndtv[i])
    if abs(ans*10)<1:
        ans=ans*10
    dic_ndtv_score[i]=ans

dic_ndtv_score
```

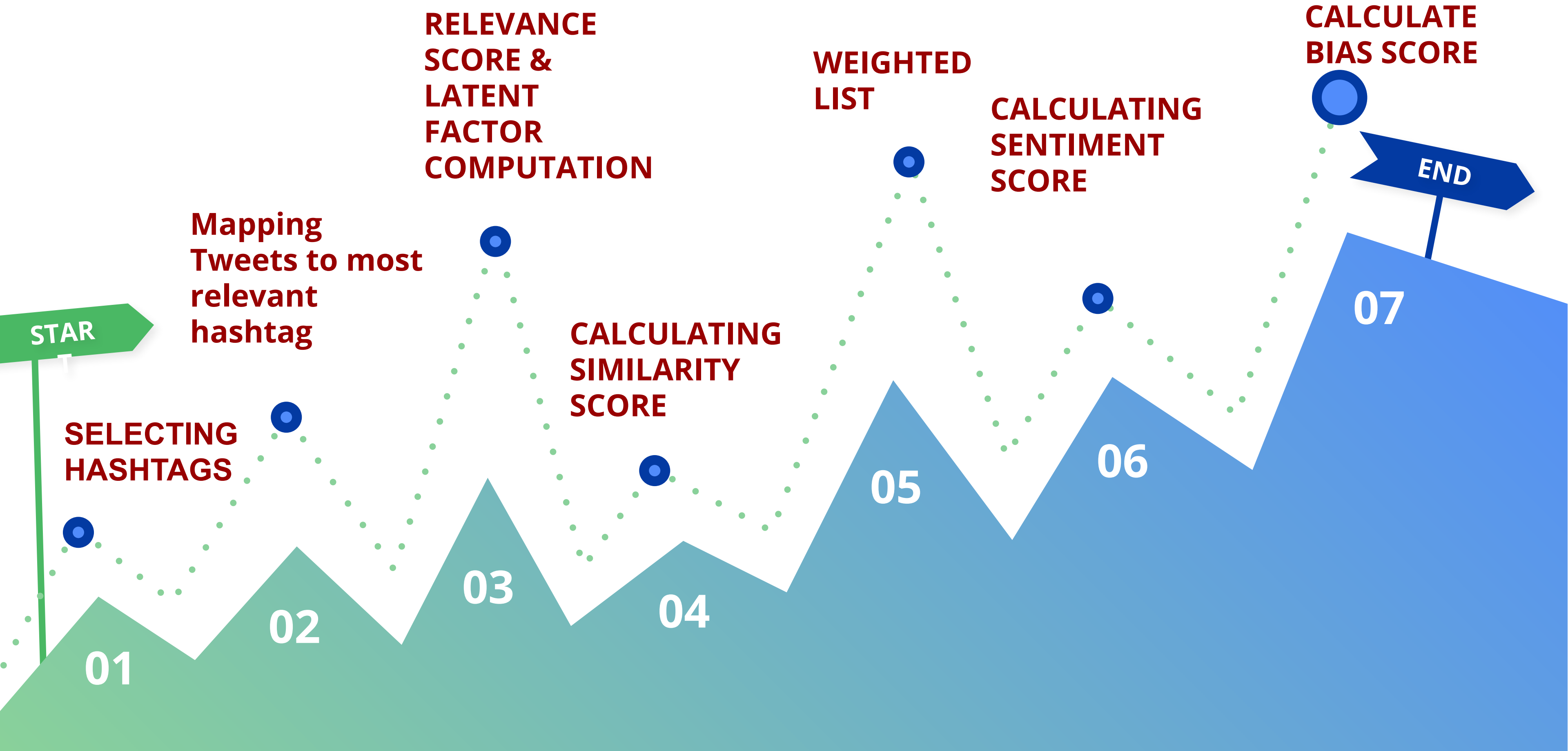
```
{'abhinandan': -0.15593333333333334,
 'modi': 0.5230416866315271,
 'elections': 0.3150411599625818,
 'worldcup': 0.6297997799779984,
 'budget2019': 0.12868382352941193,
 'congress': -0.12598498659517424,
 'bjp': 0.17921435594886898,
 'politics': 0.17269791666666662,
 'mumbairain': -0.4489452054794521,
 'kashmir': -0.19661208333333338}
```

2 ZEE

```
dic_zee_score={}

for i in dic_zee.keys():
    total_score=0.0
    for j in dic_zee[i]:
        total_score+=sentiment_analyzer_scores(j)
    l=(len(dic_zee[i]))
    ans=total_score
    if l !=0 :
        ans=ans/len(dic_zee[i])
    if abs(ans*10)<1:
        ans=ans*10
    dic_zee_score[i]=ans
dic_zee_score
```

```
{'abhinandan': 0.1217171428571429,
 'modi': 0.2827373211963589,
 'elections': 0.46303914590747264,
 'worldcup': 0.12076774193548385,
 'budget2019': 0.7787058823529411,
 'congress': -0.49409905020352746,
 'bjp': 0.15343795036028815,
 'politics': -0.24791549295774656,
 'mumbairain': -0.9246571428571431,
 'kashmir': -0.1814639175257732}
```

7 Step Mountain Journey to map each tweet with a hashtag
And finding the bias



$$FinalBias(c,p,H) = \frac{\sum_{i=1}^{m} Bias(c,p,h_i)}{m}$$

Baising Score

Baising Score of a hashtag = similarity score of that hashtag* sentimental score of that hashtag

1 NDTV

In [51]:
```
dict_baised_ndtv_bjp={}
for i in list(dict_sim_ndtv_bjp.keys()):
    dict_baised_ndtv_bjp[i]=(dict_sim_ndtv_bjp[i]*dic_ndtv_score[i])
```

In [52]:
```
dict_baised_ndtv_bjp
```

Out[52]:
```
{'abhinandan': -0.05722749943823471,
 'modi': 0.2390658238184127,
 'elections': 0.10086728182526994,
 'worldcup': 0.017275262118066872,
 'budget2019': 0.013248783846851336,
 'congress': -0.04033683455887994,
 'bjp': 0.052650728890043214,
 'politics': 0.08958534394370388,
 'mumbairain': -0.046094316573904295,
 'kashmir': -0.0898651692132728}
```

In [53]:
```
print('Baising Score of ndtv in respect of bjp' ,numpy.average(list(dict_baised_ndtv_bjp.values())))
```

```
Baising Score of ndtv in respect of bjp 0.027916940465805617
```

```python
dict_baised_ndtv_congress={}
for i in list(dict_sim_ndtv_congress.keys()):
    dict_baised_ndtv_congress[i]=(dict_sim_ndtv_congress[i]*dic_ndtv_score[i])
```

```python
dict_baised_ndtv_congress
```

```
{'abhinandan': -0.013975028643581737,
 'modi': 0.20542915321073194,
 'elections': 0.17670008587417868,
 'worldcup': 0.19596023310560828,
 'budget2019': 0.04104966326275641,
 'congress': -0.02725806478251636З,
 'bjp': 0.11077883326819557,
 'politics': 0.07338312224734295,
 'mumbairain': -0.11600286335929785,
 'kashmir': -0.04253891713102577}
```

```python
print('Baising Score of ndtv in respect of congress' ,numpy.average(list(dict_baised_ndtv_congress.values())))
```

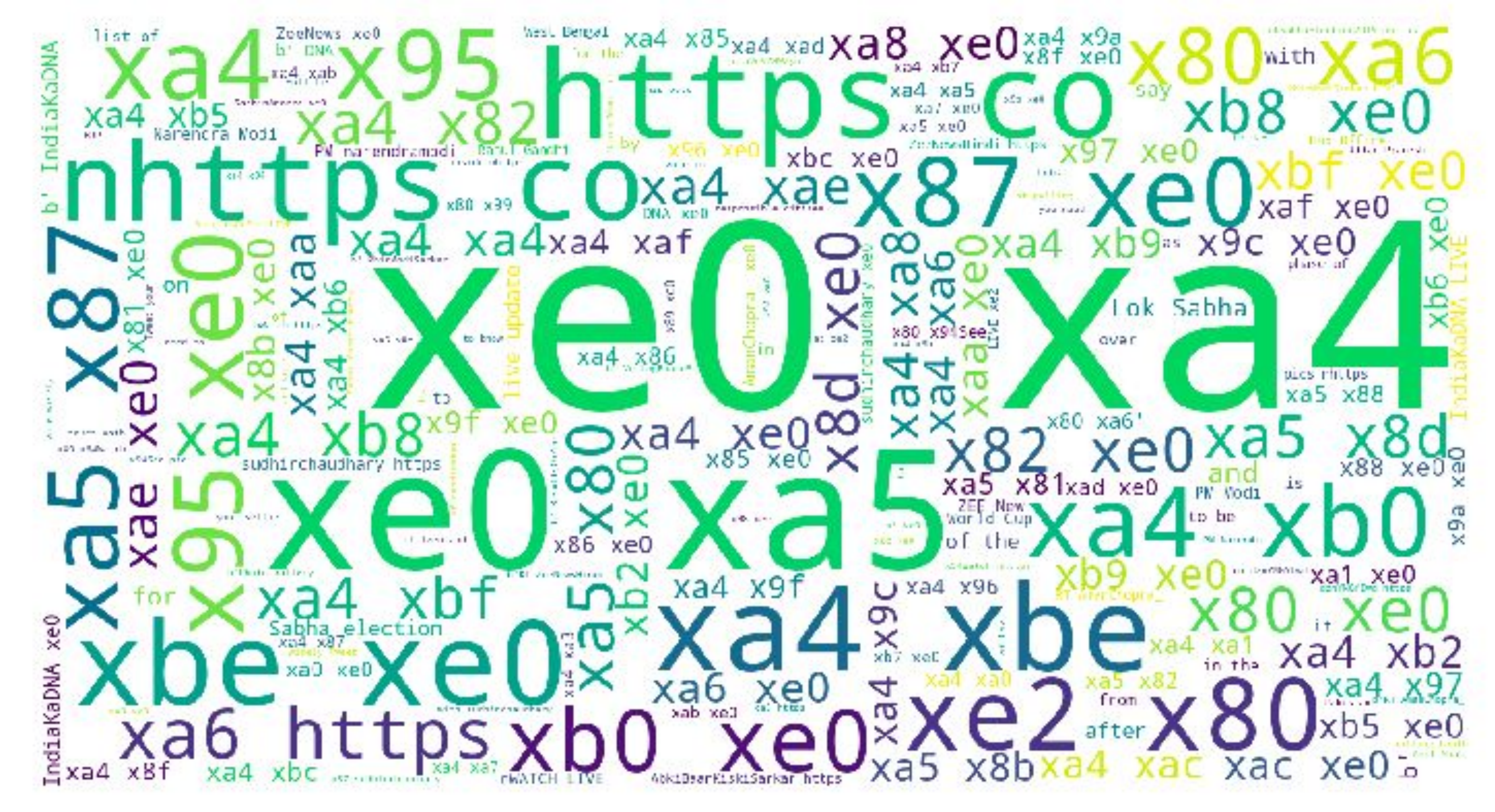
```
Baising Score of ndtv in respect of congress 0.060352621705239216
```

Final Biasing scores for each News channel for both the Political Parties-

NDTV	-	BJP	-	0.0279169404658056 17
NDTV	-	Congress	-	0.060352621705239216
ZEE	-	BJP	-	0.09268904734575159
ZEE	-	Congress	-	0.017833281510493064
Republic	-	BJP	-	-0.014459628126133165
Republic	-	Congress	-	-0.005504143535630761

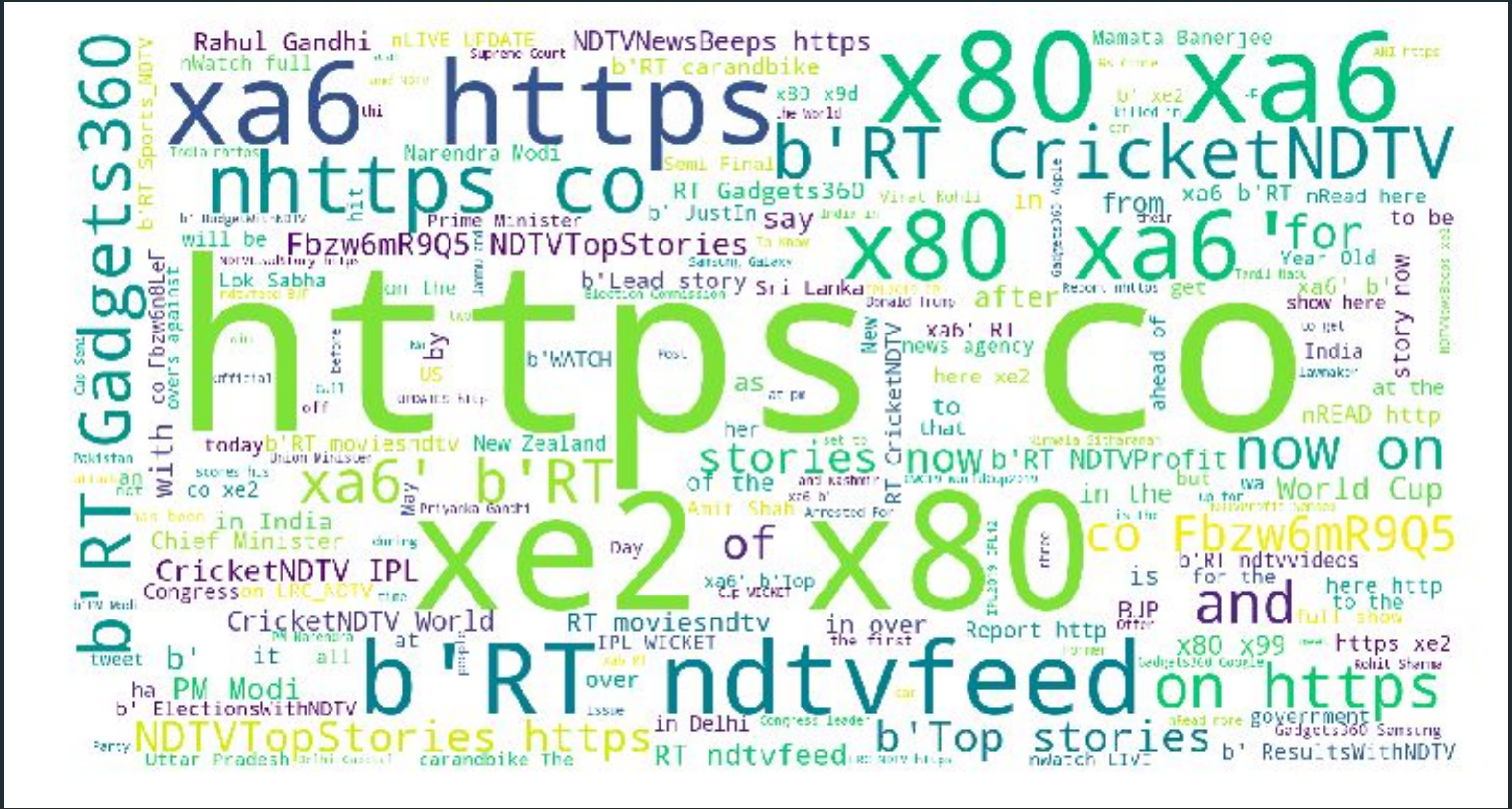
ADDITIONAL

Word Cloud for ZEE News



Word Cloud for Republic

Word Cloud for NDTV



References

[1] Wei Feng , Jianyong Wang ,"We Can Learn Your #Hashtags: Connecting Tweets to Explicit Topics" in WWW ,2013

[2] Anand, B.; Di Tella, R.; and Galetovic, A. 2007.Information or opinion? media bias as product differentiation.Journal of Economics & Management Strategy 16(3):635–682.

[3] Bernhardt, D.; Krasa, S.; and Polborn, M. 2008.Political polarization and the electoral effects of media bias.Journal of Public Economics 92(5-6):1092–1104.

[4] Eberl, J.-M.; Boomgaarden, H. G.; and Wagner, M. 2017.One bias fits all? three types of media bias and their effects on party preferences.Communication Research 44(8):1125–1148.

[5] Hamborg, F.; Meuschke, N.; and Gipp, B. 2018.Bias-aware news analysis using matrix-based news aggregation.International Journal on Digital Libraries 1–19.

[6]https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-3-zipfs-law-data-visualisation-fc9eadda71e7

[7]https://medium.freecodecamp.org/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3

[8]https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

[9]https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/

THANK YOU