# Impact Measurement of Tweets

**Mert Koşan** mertkosan@sabanciuniv.edu
*Computer Science and Engineering, Senior*

**Abdullah Usame Günay** abdullahgunay@sabanciuniv.edu
*Computer Science and Engineering, Junior*

**Egemen Yiğit Kömürcü** ekomurcu@sabanciuniv.edu
*Computer Science and Engineering, Junior*

**Inanç Arın** inanarin@sabanciuniv.edu
*Computer Science and Engineering, Supervisor*

**Yücel Saygın** ysaygin@sabanciuniv.edu
*Computer Science and Engineering, Co-supervisor*

## Abstract

This research project is focusing on the impact of tweets on Twitter. There are some factors in Twitter -retweets, likes or quotes- that can show the impact. However, this is giving missing information about the impact of tweets. People can copy or change tweet a little bit with some comment and post as her/his tweet. We are calling it “Hidden Retweets” by which could help to get more accurate analysis performance. Also during the process of finding hidden retweets, we need to define some threshold values and make preprocessing to increase accuracy and speed of the process.


**Keywords:** hidden retweets, threshold, general tweets, sensitivity analysis

## 1. Introduction

Twitter is a social media platform which people can share their ideas or information about any topics or specific tweets. Every second averagely 6000 tweets are posting on Twitter (Internet Live Stats, 2018) and this corresponds about 500 million tweets are tweeted per day. This is really huge and valuable data. We believed that this information is worth research. People are tweeting every second, but what is the impact of these tweets? In this report, we will talk about the impact of tweets in main two questions 1) How can we measure the impact of tweets? and 2) During the measurement how should we decide the threshold value of parameters? In Twitter, there are obvious criteria that show the impact of tweets, which could be the number of likes, retweets or quotes. However, we believed these criteria cannot show the real exact impact of tweets. This is because people could copy someone's tweet as her/his tweets or they can change a little bit information on that tweet before posting it. This kind of tweets can be defined as "Hidden Retweets" (Arin, 2017). However, the definition of Hidden Retweets are not really clear and we will try to extend it or make it more robust. For this aim, we will talk about our experiments and methods that can be used.

Our source of inspiration is Ph.D. Thesis, Impact Assessment & Prediction of Tweets and Topics by Inanç Arın (2017). In that thesis, experiments and analysis are made with specific topics, whereas we will make our analysis on all tweets that are collected, so we are not making narrow the tweets. In the thesis, there are many methods which are deciding similarity value between tweets. In this report, we will talk about some of these methods and we will propose threshold value for this similarity value and make preprocessing to increase the performance of analysis.

One of the main problem in this research is finding similarity values between tweets and this is a very complex problem. We tried to choose an efficient and suitable algorithm and data structures. Longest Common Subsequence is a good algorithm to find similarity between two strings however, it has high time complexity for our problem. Another method which is efficient in string operation is using Suffix Tree as a data structure.

In this project, we will specify three important steps to find out the impact of tweets. First one is about threshold value for similarity between tweets to identify hidden retweets of tweets. For this problem, we will propose sensitivity analysis. Secondly, owing to the fact that working on whole tweets is highly complex, preprocessing should be done on tweets. To satisfy this condition, we will try to eliminate tweets that could be considered as a general. Removing tweets that are sending every day -for instance "Merhaba" (Hello in English)- may increase the performance of analysis. That's why we made a statistical analysis to identify general words. The third one is related to the existence of hidden retweets. Some tweet may not have its hidden retweets, so we should not analyze this kind of tweets. Besides these three important matters, there are some factors are worth research. We won't talk about in detail in this report, but we believe that they could affect the performance. One of these is tweeting time. When we are trying to find hidden retweet of some tweet, should we consider posting time difference between these two tweets? This time can give an important clue about hidden retweets. Another problem is a length of the tweets. Short tweets are hard to analyze, so we can remove them from analyze. However, how can we decide how short tweet is?

In a short time, we collected approximately 23 million tweets and we still are collecting. We didn't give any restriction while we are collecting. At the beginning, we were collecting only text information of

tweets. However, we started to collect every information about tweets, because to make extensive analysis and obtain useful information, we need every kind of information about tweets, for example, “Is it retweet?” or “How many retweets have tweet got?.

The rest of the report is organized as follows. We will first discuss what technologies, algorithms and data structures we can use in section 2. Material and Methods. Important three steps and their details will be provided in section 3. Impacts of Tweets. In section 4. Conclusion and Future works, we will conclude and talk about future work can and will be done.

## 2. Materials and Methods

In this section, we will talk about technologies, algorithms and data structures we can use during the research. All statistical analyses are made in Python 2.7 and Microsoft Excel.

### 2.1. ELK (Elasticsearch - Logstash - Kibana)

ELK stands for the programs Elasticsearch, Logstash and Kibana which work perfectly together in analyzing big data. They are all open-source and built under the same company, Elastic. ELK stack has been a very popular application in solving data related problems due to its speed and advantage over standard SQL queries.

Elasticsearch is a very powerful tool to solve data extraction problems. It is also much faster than SQL queries which makes it a great solution for full text searching. It also interacts with RESTful web services, so it can access a set of data using an API. For our project, we needed a searching tool which also allows connection to APIs, therefore Elasticsearch was preferable.

Logstash, a data processing pipeline that allows the user to collect, process, and load data into many different sources. We collected the tweets via Logstash and transferred them to Elasticsearch in this project.

Kibana is an open-source data exploration and visualization tool that allows users to analyze the data in a well-ordered manner and clarity. It allowed us to visualize some statistics about our collected data which includes millions of tweets.

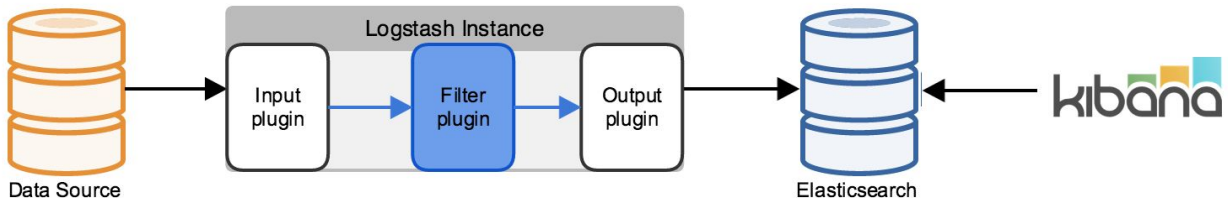



**Figure 1.** Schematic of ELK stack retrieved from https://fabianlee.org

## 2.2. Twitter Streaming API

Application Programming Interface, shortly API, is the interaction tool between web services and computer programs. Most web services allows developers to interact with the server and hence, access the data. For this project, we used Twitter's streaming API to access the tweets.

We will use Logstash's Twitter input plugin to import the tweets. Firstly, we need a configuration file which has four pieces of information to use Twitter Streaming API: consumer key, consumer secret, access token and access token secret. After getting these, we can also specify a language or a keyword to access certain types of tweets. The input field will be ready after completing these steps. We will also configure the output plugin as elasticsearch to store the logs in Elasticsearch.

## 2.3. Algorithms and Data Structures

### 2.3.1. Longest Common Subsequence

Given two strings A and B, Longest Common Subsequence (LCS) is the longest sequence of characters that appear left-to-right in A and B. It is solved with a typical dynamic programming approach.

$$L[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ L[i-1, j-1] + 1 & \text{if } A[i] = B[j] \\ \max(L[i-1, j], L[i, j-1]) & \text{otherwise} \end{cases}$$

Let L be the matrix with the dimensions (m+1) x (n+1), its elements can be computed by the above definition. The entry L[m,n] holds the length of the LCS for the strings A and B. This solution has the time complexity of O(m*n), given the sizes m and n for the string A and B.

By using LCS, we can find the similarities between two tweets, however it is a very costly operation since the running time of the algorithm will be the multiplication of the sizes of tweets. Instead, we looked into suffix tree data structure which allows a faster processing for finding similarities in texts.

### 2.3.2. Suffix Tree

A suffix tree is a data structure constructed from a text whose size is a linear function of the length of the text and which can also be constructed in linear time. Suffix trees allow fast implementations for many string operations.

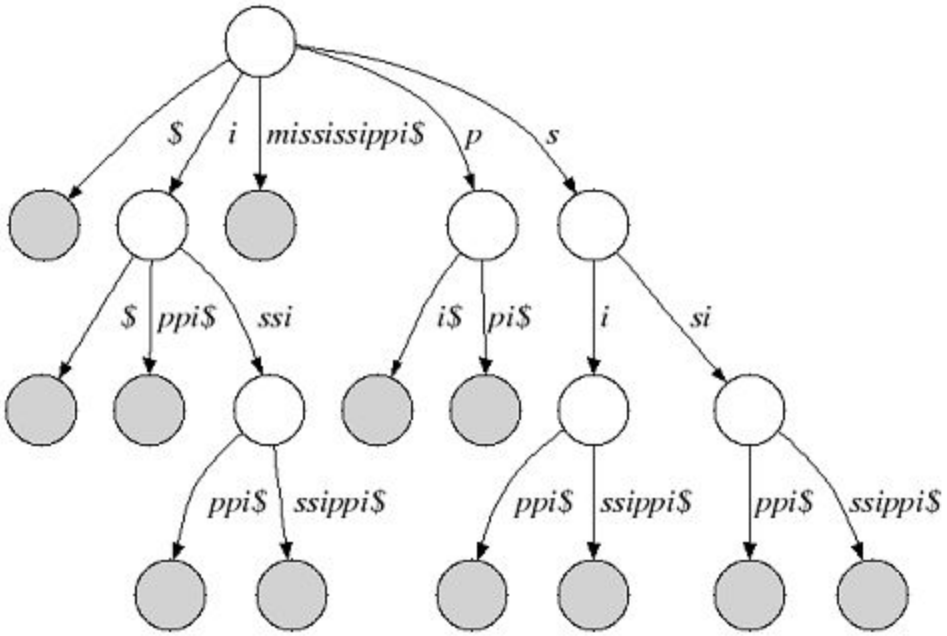



**Figure 2.** Suffix tree for the word "mississippi" retrieved from https://seqan.readthedocs.io/en/seqan-v2.0.2/Glossary/SuffixTree.html

There is an advanced version of suffix tree which is called Generalized Suffix Tree. A GST is a suffix tree for a set of strings. It is used for some operations on the strings in the set such as finding patterns, longest repeated substrings and longest palindromic substring. GST is an appropriate solution for our project since we can treat the tweets as the strings in the same set and work on that set.

The usage of suffix trees in this project is that it allows a fast searching using lexical similarity of tweets to find the hidden retweets. Assuming two tweets X and Y, since time complexity for construction of suffix trees for these tweets will be O(X) and O(Y) respectively, the generalized suffix tree of the set containing X and Y will take O(X+Y) time. It is a faster approach than LCS which would run in O(X*Y) for same operation.

## 3. Impacts of Tweets

In the process of finding hidden retweets, there are some preprocessing tasks and threshold values which should be decided. We believed that these methods will help to find more accurate results. There are three main steps that are followed during this process. In the following subsections, they will be introduced in detail.

### 3.1. Deciding Thresholds

Deciding thresholds includes two main topics of scoring the similarity values and the sensitivity analysis.

### 3.1.1 How to Score Similarity Values

First of all, similarity value in range of 0 to 1 between two tweets is calculated upon basically two algorithms stated above. Firstly, LCS(Longest Common Subsequence) is used for determining the score of the similarity value by the formula stated below:

$$normalized\ score(t_i,\ t_j) = \frac{2*LCS(t_i,t_j)}{length(t_i)+length(t_j)}$$

where i and j are the indexes of the tweet t.

On the other side, if it is desired to get benefit from suffix tree algorithm, the similarity value is calculated by the formula stated below where t is the every single tweet and v is the every node in the suffix tree . As a result, it can be achieved the time complexity of O(m+n) whilst LCS does in O(m*n) time in which m and n are the length of the two tweets.

$$score = \frac{nodelength_v}{|t|}$$

### 3.1.2 Sensitivity Analysis

Sensitivity analysis is the procedure of examining how the variation in output of a model can be apportioned to the different sources of variation in its inputs. It also provides a robustness check for the model since it translates the range of input parameters into the model and then, into the output variables.

Saltelli(2002) defines sensitivity analysis as:

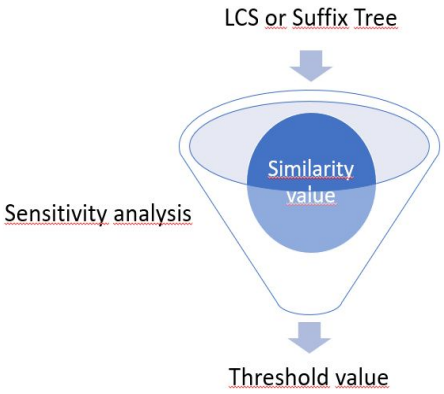
*"The study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input."*

Sensitivity analysis is widely used in decision-making process of business and economic models as well as engineering applications. The econometrician Edward Leamer claims that all scientific disciplines and especially economics need strong inferences based on organized sensitivity analyses(Leamer, 1985). Hermeling and Mennel present a formalisation of how sensitivity analyses can be implemented for economic simulations for validity of the results taken from the model(Hermeling & Mennel, 2018).

Sensitivity analysis can also be applied to some simulations for real life problems. In their paper, Charitos and van der Gaag investigate the effect of inaccuracies in threshold decision-making of parameters of a dynamic Bayesian network as an illustration of real life infectious disease (Charitos & van der Gaag, 2012).

We are finding a similarity score between two tweets, but how can we say that tweets are hidden tweets given tweet? In order to say that the tweet may be hidden retweet or not, it is required to set a threshold value for each query. In detailed, sensitivity analysis method is used to decide threshold from retrieved tweets.

Especially, we use sensitivity analysis by giving the input of the similarity value of all tweets retrieved and the output threshold is determined dynamically for each query. Consequently, after scoring the similarity values of between all tweets and written query by specific algorithms, scores are put into a model using the sensitivity analysis method to decide the threshold value that it can be said which tweets might be hidden retweets or not. Below figure represents the flowchart of whole process.




Before we apply sensitivity analysis, we eliminate some tweets with similarity score smaller than 0.69. We decided this value from 0.5 sigma value of normal distribution. We chose 0.5 sigma value, because 1 sigma value (~ 0.89) could be larger for our problem.

To be able to apply sensitivity analysis to our problem, we added another parameter to the similarity score. This parameter refers to the result of LCS or Suffix Tree algorithms. We added this parameter, since in sensitivity analysis we will measure the sensitivity of a given similarity score. For each similarity score, we created a block with size of [score - 0.01, score + 0.01] to get more data between these values. For each block, we apply two sensitivity analysis method.

i) The Piecemeal Approach:

We found max difference between values in block, and this max value refers to sensitivity of this similarity score. If sensitivity is too high, we may not select this score as a threshold value. This approach is deterministic sensitivity analysis (Hermeling, 2008). Accuracy level is not good, more computationally fast.

$$\Delta = \max_{a_i, a_j \in \{a_1, \ldots, a_M\}} |h(a_i) - h(a_j)|$$

For instance, for similarity value 0.85, we created a block with range between 0.84 and 0.86. In these range, all similarity values have the result came from LCS or Suffix Tree. The maximum range from these values will assign to sensitivity to 0.85.

ii) The Monte-Carlo Approach

In this approach, we are finding variance of the block and this variance value refers to sensitivity of the similarity score. Again, higher sensitivity can tell us the similarity score is not suitable one to choose as a threshold value. According to Hermeling, this approach is stochastic and requires more computational power than The Piecemeal Approach. However, variance may show us better result.

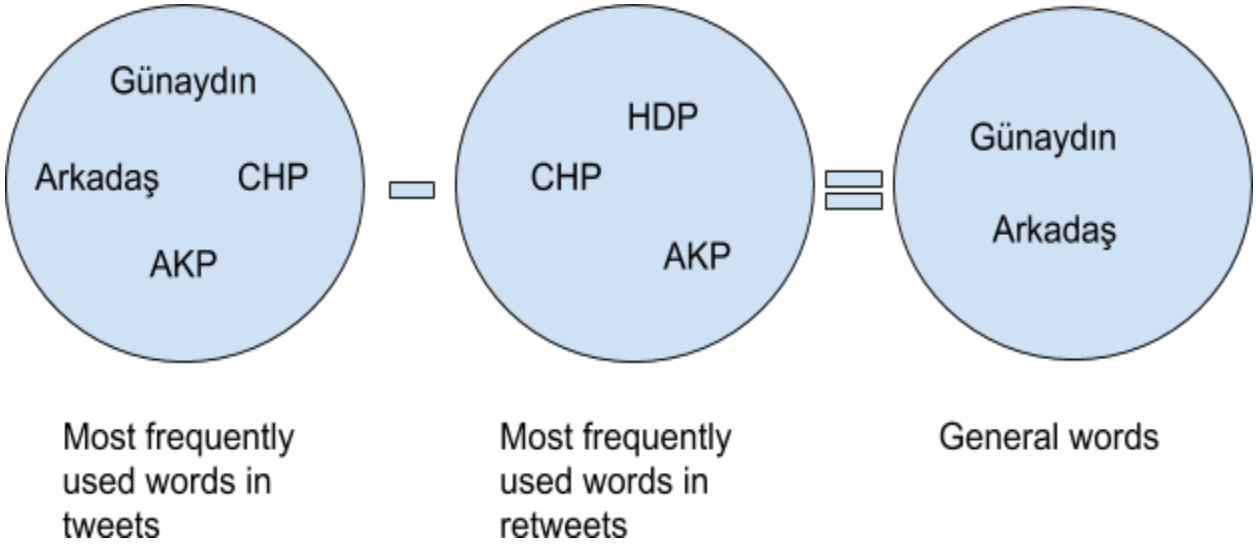
To give a general example of sensitivity analysis, if you query "Same-sex couples should be able to get married." among the retrieved billions of tweets, similarity values in range of 0 to 1 for each tweet is formed by algorithms of LCS or Suffix Tree. If three tweets has scores of 0.2, 0.4 and 0.9 and the threshold above query is determined as 0.8 through sensitivity analysis, the tweet whose similarity value is 0.9 can be considered as it is likely to be hidden retweet. Therefore, it can be stated that the impact of the queried tweet is increased. Also through sensitivity analysis, threshold is not determined as 0.8 statically as you query for "Jamie Vardy. Remember the name and the tweet." So, threshold must be determined dynamically for every query.

Deciding threshold was a hard issue in our problem. However, with sensitivity analysis, we dealt with the huge problem and identified the similar tweets to an original tweet. These similar tweets can be called as hidden retweets.

## 3.2. General Tweets

First, we need to define what general tweets are. In our concept, general tweets are tweets that are tweeted everyday. For example, "Günaydın" (Good morning in Turkish) is a common phrase and is being tweeted everyday. So when we are trying to impact of tweet, we need to eliminate these general tweets, because every general tweets can be independent from each other and they are very similar to each other. We are interested of being dependency rather than similarity.

When we see tweet, first we need to define whether that tweet is general or not. We believe that finding general words can help us to identify tweet is general or not. So, we tried to find most general words. To achieve this goal is not easy as it is seen. Firstly, we extract most frequently used words from tweets that we collected. However, it cannot be said that these words are general words, because some most frequently used words can be important for us to find hidden retweets for any tweet or topic. So, we need to determine these words and extract them from the most frequently used words set. One method to find these words can be looking most retweeted tweets (bigger than 1000 retweets) and extract most frequently used words from these tweets. Resulting set can give us general words set and we believe that we can find general tweets by using general words set. Below diagram explains what we are trying to do.




In the following subsections, we will clarify methods and process of finding most general words.

### 3.2.1. Challenges and Preprocessing

Tweets consist of many unrelated objects and mistakes that should not be considered in our analysis. So, we need to detect and eliminate these features from each tweet. The main problems with the tweets are links, mentions, and emojis. We deleted them from the text. Other problems are punctuations because when any punctuation is used after a word, it can reduce the frequency of the word. For example, “sevgi” and “sevgi(:” must be considered as the same word. So, we removed every character except letters and space.

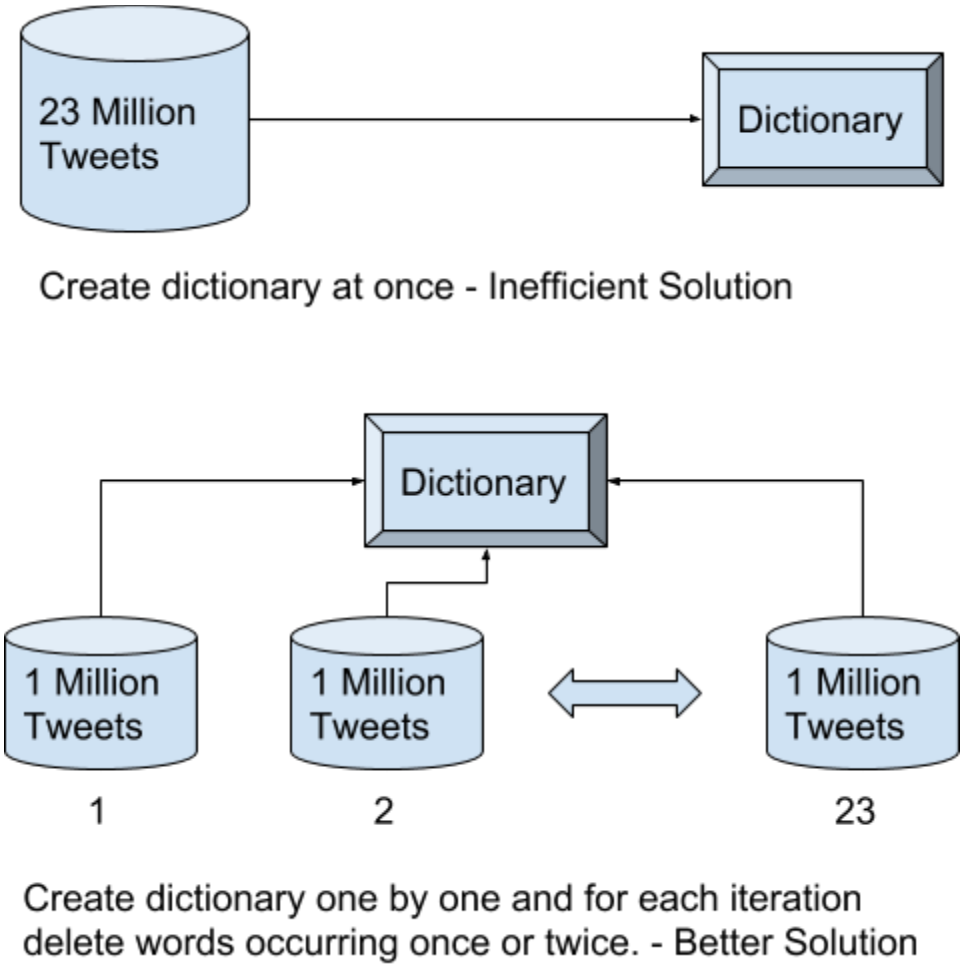
In this analysis, we used nearly 23 million Turkish tweets that approximately contains 244 million words in it. Another problem arises when Python dealing with Turkish letters. When we are creating built-in Python dictionary (dict {}), we realized that key fields of the dictionary cannot contain Turkish character. So, we first transform Turkish characters to English ones to speed up our implementation, but then we found a way to cope up with this situation.

Stop words are the really important factor in statistics. So, we found Turkish stop words (Bougé, 2011) and removed them from our dictionary. Stop words are very common words that have little value in sentences. Indexing the sentences without using stop words can increase the performance according to IBM. However, there are some cases that stop words won’t be removed (IBM, 2010).

### 3.2.2. Complexity and Statistics

We had 23 million tweets that are approximately 2 GB in a text file, so reading this file and creating a dictionary from it could be memory inefficient solution. We decided to divide 23 million tweets into 23 buckets which consist of 1 million tweets each. We assumed that if any word can occur one or two times in 1 million tweets, it can be discarded. After creating the dictionary from each 1 million, we removed one or two occurrence words from the dictionary.

This really affects our time and memory performance. We didn't load all 2GB into memory. Below schema is introducing what we did.




Next question is how we decide the most frequently used tweets. We cannot just define random threshold value. We first looked word distribution, which basically tells how many words exist for each occurrence level (three occurrences = level three, five occurrences of word = level five so on). We tried to obtain normal distribution so we can decide threshold value based on three-sigma rule of thumb (Kazmier, 2004), which says a small amount of data is gathering at three-sigma distance. Three-sigma distance is mean of data values plus three standard deviation of data values. Below figure illustrates normal distribution and sigma values.

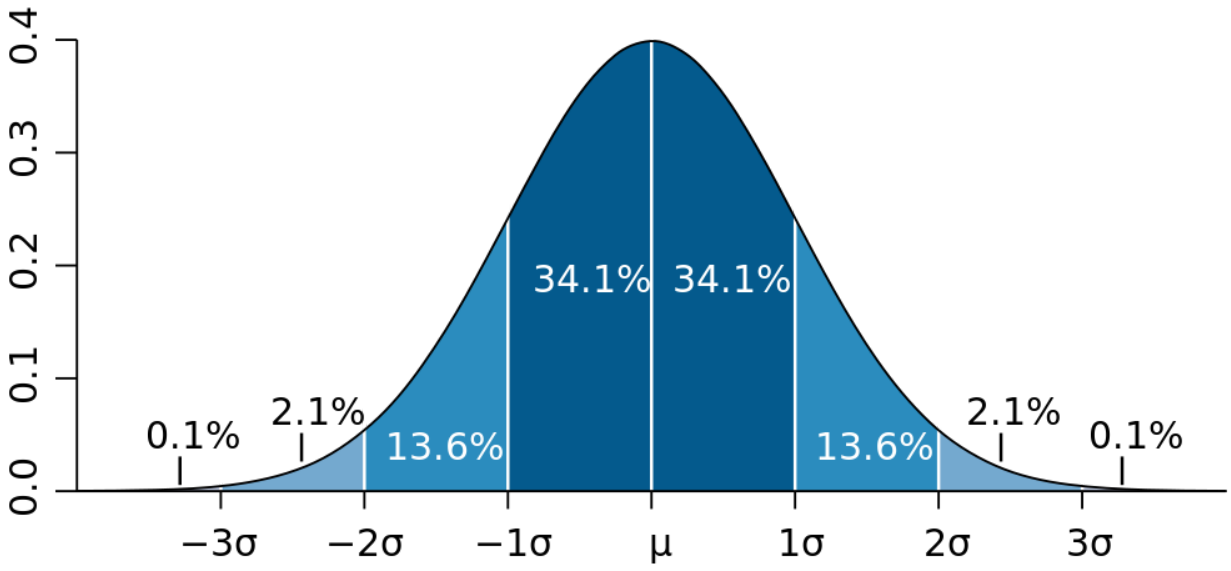



**Figure 3.** There exists 0.1% data right of the three-sigma value and 2.1% data right of the two-sigma value. retrieved from: https://thecuriousastronomer.files.wordpress.com/2014/06/1000px-standard_deviation_diagram-svg.png

Three-sigma value is a good approach for deciding threshold value; however, the distribution of our data is more like exponential decay, so we couldn't fit these data into a normal distribution. We decided to use some assumptions to create a good model for the decision of most frequently used words.

The first assumption was normal distribution assumption with two-sigma value threshold. We didn't use three-sigma value threshold because exponential decay function will lose many data with three-sigma rule. We found threshold value 21872 occurrences which mean words, in 244 million words, that are occurring more than 21872 times can be most frequently used words. However, to see robustness of this solution, we made other experiments with different assumptions. At the beginning, we assumed that first 100 most frequently used words should be included in most frequently used words set, so we removed them from our analysis. This helped us a lot because they are more like outliers in statistics.

In the experiments, we defined two parameters which are lower bound and upper bound. Words which occur less than lower bound cannot be considered as most frequently used words, so remove them and words which occur more than upper bound will be considered as most frequently used words, so remove them from the analysis. This could help us to analyze data more effectively and find useful results. To determine the lower bound, we chose k value that tells us that word should occur at least one time for each k words to be defined as most frequently used. So lower bound is defined as a total number of words divided by k value. The upper bound had two options in our experiments, first one is the occurrence count of most frequently used the word after the removal of first 100 most frequently used words. Second is the value we found by normal distribution assumption analysis, 21872.

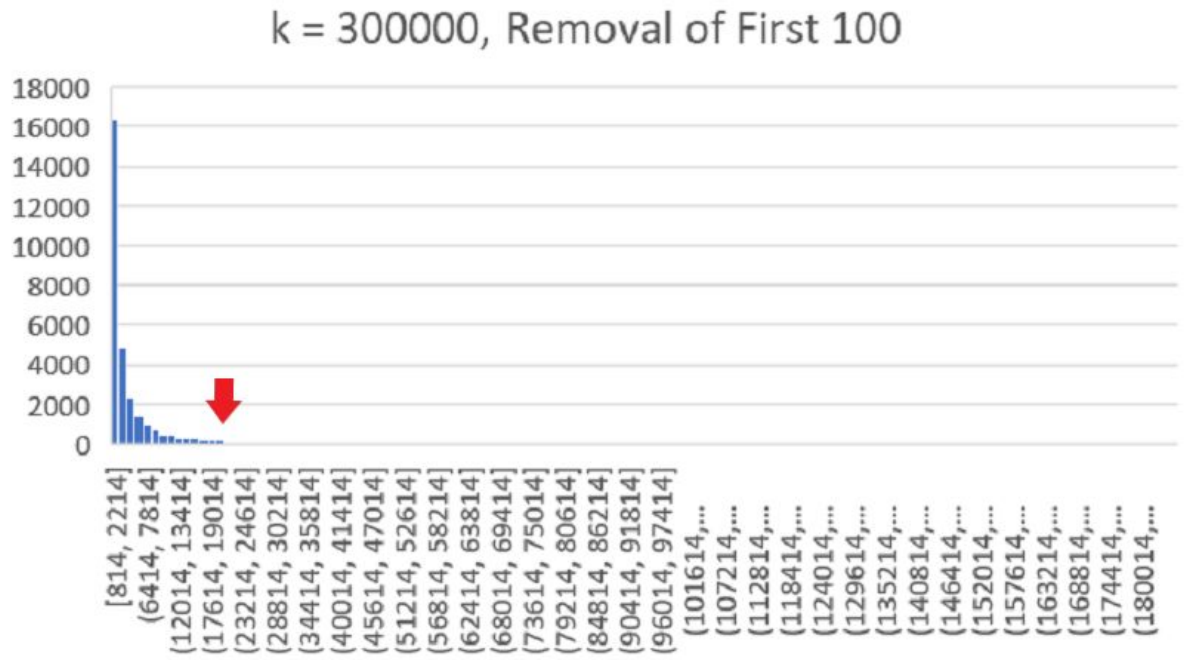
Below equations shows lower bound and upper bound values for our experiments. We have 244 million words in our analysis.

$$Lower\ Bound\ =\ \#\ Total\ Words\ /\ k$$

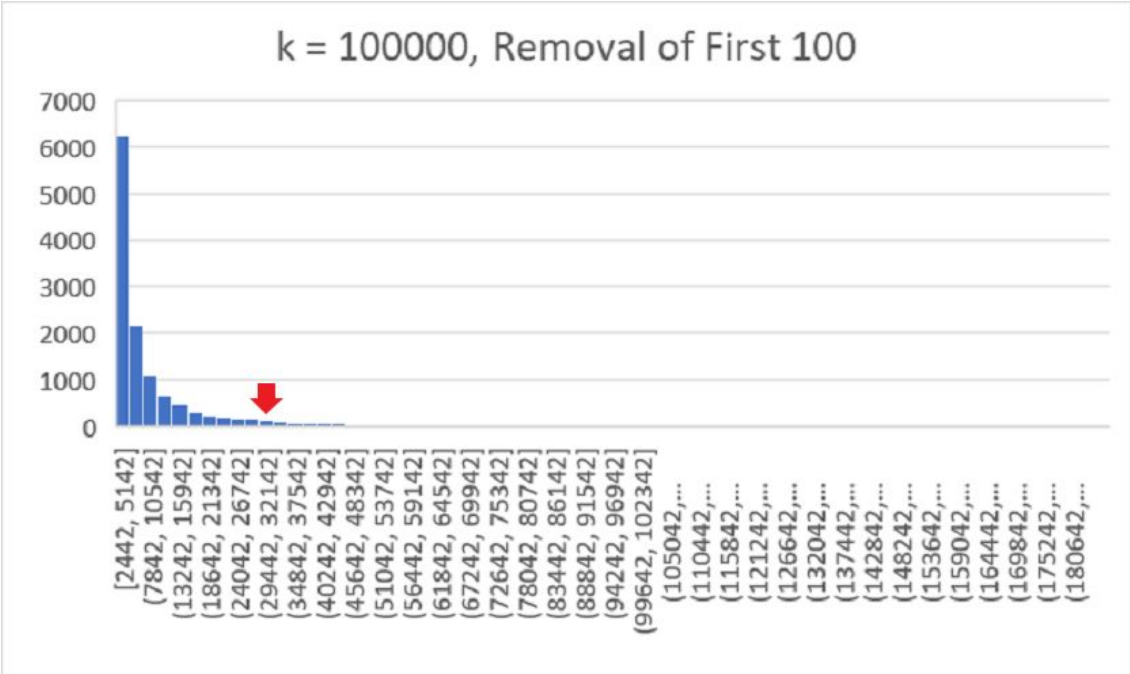
$$Upper\ Bound_1\ =\ Most\ Occurrence\ Count\ After\ Removal\ of\ First\ 100$$

$$Upper\ Bound_2\ =\ Normal\ Distribution\ Analysis,\ 21872$$

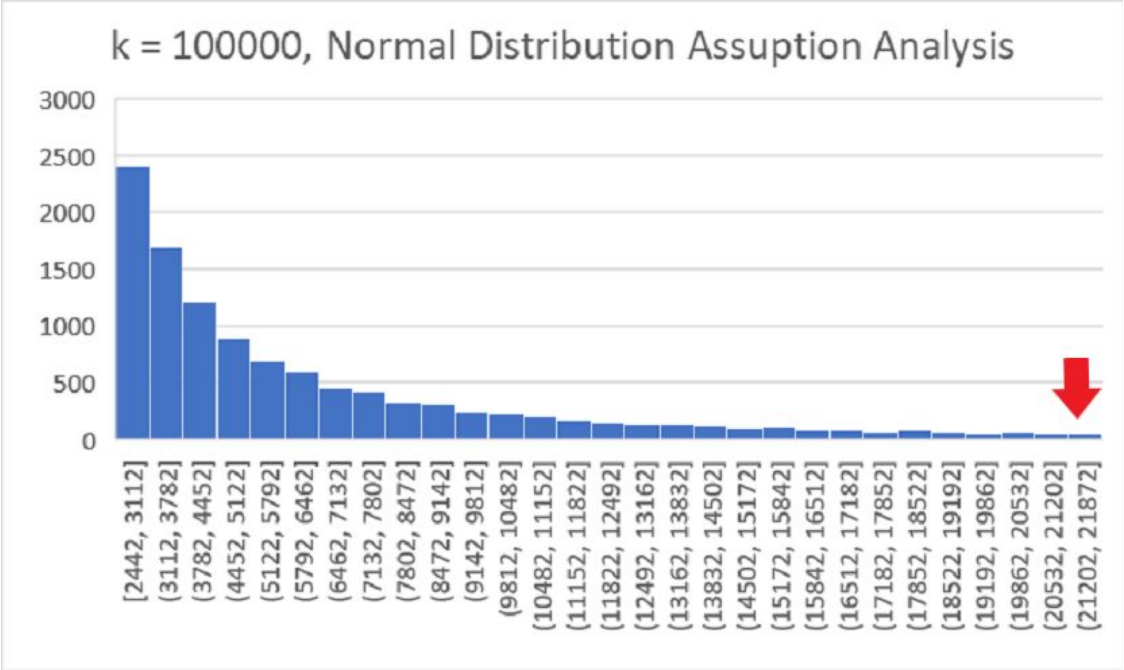
Below you can find three experiments that we made. For each experiment, we created a histogram and indicated handmade fading point of graphs with a red arrow.




In this experiment, we selected lower bound as $\frac{244\ million}{300000} \simeq 814$, so words with lower frequencies are discarded, and upper bound as 186254.




In this experiment, we selected lower bound as $\frac{244\ million}{100000} \simeq 2442$, so words with lower frequencies are discarded, and upper bound as 186254.




In this experiment, we selected lower bound as $\frac{244\ million}{100000} \simeq 2442$, so words with lower frequencies are discarded, and upper bound as 21872 got from normal distribution assumption analysis.

All these experiments show that fading point of each graph is approximately between 20000 and 30000, so normal distribution assumption analysis gave us a useful result which is 21872. So, words with frequencies above 21872 could be counted as most frequently used words in tweets with this dataset.

The next step will be found most frequently used words in retweets to obtain most general words on Twitter. However, this will not introduce in this report, this is because we didn't analyze, yet. It will be our future works.

### 3.3. Does every tweet has hidden retweets?

We also try to answer the question whether every tweet has hidden retweets or not. As Arin(2017) states, every tweet may have hidden retweets, however every tweet may not have hidden retweet. In other words, if tweets whose similarity values are above the sensible threshold, this tweet have hidden retweets which has impact on the original tweet. On the other hand. if all retrieved tweets are not similar to the original tweet, it does not have hidden retweets. For instance, if we retrieved billions of tweets whose similarity values are mostly less than or equal to 0.1, hidden retweets do not have impact on the original tweet so it can be emphasized that sensitivity analysis is not required to be done in this case.

As all aforementioned factors must be under consideration, finding hidden retweets is not an easy process.

## 4. Conclusion and Future Works

Our main aim is calculating the impact of tweets. In Twitter, we have retweets, likes and quote number for a specific tweet; however, we also consider hidden retweets (Inanc, 2017). These hidden retweets can be created by directly copying tweet or posting the same tweet with little change.

Finding these hidden retweets is a hard problem. Because we need to define similarity values between tweets and this is a complex problem. We talked about Longest Common Subsequence algorithm and Suffix Tree data structure that can be used to find this similarity. In this paper, we focused on which tweets with similarity values are hidden retweets. We believed that static threshold value won't work on this problem because our tweet data is extending day by day and it can be different for different datasets. The dynamic threshold value that is found by sensitivity analysis could be more useful. Another problem was finding general tweets, that should be discarded during the process of detecting hidden retweets. We made a statistical analysis and found threshold value that will decide most frequently used words. This could help us to find most general words. For the third step, according to Arin, every tweet doesn't have to have hidden retweets (2017). So, we claim that we need to find a distinction between tweets that have hidden retweets and that haven't.

In this report, we didn't make an experiment of combining finding similarity with given threshold values and discarding general tweets. This can be one of the future work that can be done. In section 3.2, we found most frequently used words in Twitter given dataset. However, also most frequently used words from retweets also should be found and this will lead to finding most general words that are actually ultimate goal. Turkish words contain a lot of suffixes in words, so this is really hurting the analysis of counting words. As a future work, these problems can be solved and we believed that this may boost the accuracy of result drastically.

## 5. References

Arin, Inanc. (2017), Impact Assessment & Prediction of Tweets and Topics, PhD Thesis, Sabancı University

Blumsack, S. (2017). Energy Markets, Policy, and Regulation. Retrieved from https://www.e-education.psu.edu/eme801

Bougé, K. (2011), Turkish stop words, retrieved from https://sites.google.com/site/kevinbouge/stopwords-lists

Charitos, T., van der Gaag, L. (2012). Sensitivity Analysis for Threshold Decision Making with Dynamic Networks

Hermeling, C., Mennel, T. (2008). Sensitivity analysis in economic simulations - a systematic approach. Center for European Economic Research

IBM (2010). Stop Word Removal, IBM Content Analytics, Version 2.2.+ retrieved from https://www.ibm.com/support/knowledgecenter/en/SS5RWK_2.2.0/com.ibm.discovery.es.ta.doc/iiysalgstopwd.htm

Internet Live Stats. Twitter usage statistics, 2018. retrieved from http://www.internetlivestats.com/twitter-statistics/.

Kazmier, Leonard J. (2004). Schaum's Outline of Business Statistics. McGraw Hill Professional, 4th edition

Leamer, E. (1985), "Sensitivity Analyses would Help", American Economic Review 75 (3), 308-313

Saltelli A. (2002). Sensitivity Analysis for Importance Assessment, Risk Analysis, 22 (3), 1-12.