6-1-2016

Trending topic extraction from social media

Nada Ayman A. Mostafa

The American University in Cairo

School of Science and Engineering

TRENDING TOPIC EXTRACTION FROM SOCIAL MEDIA

A Thesis Submitted to the Department of Computer Science and Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science

By

Nada Ayman A. Mostafa

Under the Supervision of

Prof. Dr. Ahmed Rafea

Spring 2016

DEDICATION

My parents, my husband and my lovely kids

Kids, remember that patience and persistence are the main keys to success

Love you all

ACKNOWLEDGEMENT

I'm very grateful to God that I could reach this point and for that I'd like to thank Dr Ahmed Rafea for his continuous guidance and support and for answering all my questions without delay or hesitations. It was an honor and a blessing working with such a great professor.

I would also express my thanks to Dr Sherif Aly who has been a great support since I joined the university, and Dr Amr El Kadi for his suggestions and advices.

As this thesis was a part of a bigger project called "Semantic Analysis and Opinion Mining for Arabic Web" funded by ITIDA , I would like to recognize their role in this research. Also the team of the Egyptian industrial company LINK-development for their help in the research.

Finally I'm very grateful to all my family who supported me and kept pushing me to reach this phase, for believing in me and never letting me down. My parents, I can never be thankful enough for everything they have done for me. My late grandparents whom I wished they could witness this moment. Last but not least my dear husband who was always a great support to me at all times, I couldn't have done it without him being there for me.

ABSTRACT

Social media has become the first source of information for many people. The amount of information posted on social media daily has become very vast that it became difficult to track. One of the most popular social media applications is Twitter. Users follow lots of news accounts, public figures, and their friends so they can be updated by the latest events around them. Since the dialect language and the style of writing differ from a region to another, our objective in this research is to extract trending topics for an Egyptian twitter user. In this way, the user can easily get at a glimpse of the trending topics discussed by the people he follows. To find the best approach achieving our objective, we investigate the document pivot and the feature pivot approaches. By applying the document pivot approach on the baseline data using tf-itf (term frequency-inverse tweet frequency) representation, repeated bisecting k-means clustering technique and extracting most frequent n-grams from each cluster we could achieve a recall value of 100% and F1 measure of 0.8. The application of the feature pivot approach on the baseline data using the content similarity algorithm to group related unigrams together, could achieve a recall value of 100% and F1 measure of 0.923. To validate our results we collected 12 different data sets of different sizes (200, 400, 600, and 1200) and from three different domains (sports, entertainment, and news) then applied both approaches to them. The average recall, precision and F1 measure values resulted from applying the feature pivot approach are larger than those achieved by applying the document pivot approach. To make sure this difference in results is statistically significant we applied the Two-sample one-tailed paired significance t-test that showed the results are significantly better at confidence interval of 90%

The results showed that the document pivot approach could extract the trending topics for an Egyptian twitter user with an average recall value of 0.714, average precision value of 0.521, and average F1 measure value of 0.556 versus average recall, precision and F1 measure values of 0.981, 0.754, and 0.833 respectively, when applying the feature pivot approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

rb	**repeated bisecting k-means clustering technique**
agglo	**agglomerative clustering technique**
bagglo	**biased agglomerative clustering technique**
tf-itf	**term frequency-inverse tweet frequency**
K	**number of clusters**
KEA	**key-phrase extraction algorithm**
FCU	**frequency common unigrams**
Bi30	**bigrams occurring more than 30% of the cluster size**
Bi25	**bigrams occurring more than 25% of the cluster size**
Bi50	**bigrams occurring more than 50% of the cluster size**
Uni25	**nigrams occurring more than 25% of the cluster size**
Uni30	**unigrams occurring more than 30% of the cluster size**
Uni50	**unigrams occurring more than 50% of the cluster size**
Tri25	**trigrams occurring more than 25% of the cluster size**
Tri30	**trigrams occurring more than 30% of the cluster size**
Tri50	**trigrams occurring more than 50% of the cluster size**

Chapter 1. Introduction

In this chapter we discuss the problem definition then we present the background of the idea of topic detection and extraction. In the third section we state research questions proposed in the objective and how we will answer these questions in the methodology section.Finally the thesis layout is presented.

1.1. Problem Definition

Over the past few years the social media has become the new social life. People share their interests, favorite places, their thoughts, and opinions about almost everything. People communicate via social media now more than they do in real life.

The pervasiveness of the social media made it easier for people to post anything at anytime from anywhere. It became the new source of news as it offers real time up to date events reporting. The Arab Spring, or presidents tweeting and posting messages on Facebook and Twitter instead of using official public media are examples of how influential social networks have become. (Rosa et al, 2014)

Twitter is a popular micro blogging service that enables users to send and read short text messages. It was launched on July 2006; monthly active users in December 2015 were estimated to be 320 million worldwide. With 80% of the users use twitter from their mobile phones, Twitter has become a part of people's lives. (https://about.twitter.com/company)

Twitter users follow news media, and public figures to keep track of events happening all over the world. They also follow people with similar interests and their friends. With the massive amount of events and information posted every day on twitter, it became more difficult to keep track of all events happening.

News spread way faster and more effective through social media. Due to the real time nature of Twitter, the event can be posted once it happens before being published in newspapers or even stated on TV. Twitter doesn't rely on reporters like traditional news media, anyone can post anything and it can go viral in no time. Twitter today is becoming a standard domain for event

detection, it can be used as a sensor to gather up to date information about the state of the world. (Petrovic et al, 2013). Almost all the mass media (newspapers, TV, radio stations) recently have accounts on Twitter and post news as Tweets once they happen even before they do in their usual media.

With the massive posts about different topics, it can be hard for the user to know all the events happened in a specific time period, without going through all the posted tweets in that period. Grouping tweets about the same topic and label them, can make it easier for the user to easily access tweets about a certain topic.

Twitter grows very fast which makes it harder for this task to be done manually. The existing trending topics option in Twitter shows the top 10 hash tags per specific region not per user. Our research focuses on the user's personal interests so it extracts the trending topics for a Twitter user.

1.2. Background

The idea of this research domain has originated back in the 1990's with a project called TDT (Topic Detection and Tracking). The basic idea originated in 1996, when the Defense Advanced Research Projects Agency (DARPA) realized it needed technology to determine the topical structure of news streams without human intervention (Allan et al, 1998). Topic detection is the problem of identifying stories in several continuous news streams that pertain to new or previously unidentified events. It involves detecting the occurrence of a new event such as a plane crash, a murder, a jury trial result, or a political scandal in a stream of news stories from multiple sources. Topic tracking is the process of monitoring a stream of news stories to find those that track (or discuss) the same event as one specified by a user.

Topic Detection and Tracking aims extracting topics from a stream of textual information sources and quantifying their trend in time. In general topic detection and extraction can be done using two approaches: either the documents in the collection are clustered or the most important terms are selected and then clustered. In the first method, referred as document-pivot a topic is

represented by a group of documents, whereas in the latter, referred to as feature-pivot, a group of terms describing the topic is produced instead. (Aiello et al, 2013)

1.3. Objective

The objective of this research is to identify an efficient technique for detecting and extracting trending topics for Arabic twitter user within a specific period of time.

In order to achieve this objective a set of research questions were proposed:

1. Will using the document-pivot approach lead to efficiently extracting the trending topics?
    a. Will the used clustering technique have an impact on the extracted trending topics?
    b. Will the features used in clustering affect the trending topic extracted?
    c. Will the used method of extracting the trending topic have an impact on the results?
2. Will using the feature Pivot approach lead to better extraction of the trending topics?
    a. Will different values of a threshold determining that two features related to the same topic affect the extracted trending topics?
    b. Will different values of a second threshold determining if further features related to the same topic affect the extracted trending topics?
3. Will one of the approaches give a significant difference in the results when applied on different data sizes from different domains?

1.4. Methodology

The methodology proposed to answer the first research question is as follows:

- Build a baseline system using the document pivot approach following these steps:
    - Collect data from Twitter, then annotate each tweet with its topic, and preprocess all collected tweets.

    - Represent tweets using a representation method, and cluster them using a clustering technique.
    - Evaluate the clustered topics against the topics identified from the annotated topics of the tweets.
    - Extract from each cluster the most frequent hash-tags to represent the trending topics.
    - Evaluate the extracted trending topics using hash-tags against the trending topics identified from the annotated topics of the tweets.
- Apply different clustering techniques and compare the result of each technique against the baseline results to answer the research question number 1.a.
- Replace the clustering technique used in the base line with the best one found in the previous step, represent the tweets using different features, and compare the results against the system that uses the baseline features to answer the research question number 1.b.
- Change the method of extracting trending topics using n-grams extracted from each cluster, and then compare the results against the system with the best clustering technique, best clustering features , and trending topic extraction method used in the baseline to answer the research question number 1.c

Secondly we will investigate the impact of applying feature-pivot approach to answer the second research question by doing the following:

- Extract trending unigrams (keywords) and cluster them based on two levels of content similarity to represent trending topics.
- Use different values of the threshold that determines if two trending unigrams belong to the same topic (first level of content similarity) and compare the results against the annotated data to answer the research question number 2.a
- Use different values of the second threshold that determines if further trending unigrams belong to the same topic (second level of content similarity) and compare the results against the annotated data to answer the research question number 2.b

Finally to validate our results we will apply both approaches on different sizes of data from different domains and apply the Two-sample paired t-test on the results achieved by both approaches to answer the research question number 3

1.5. Thesis layout

The rest of this document is organized as follows: Chapter 2 reviews the approaches covered in the literature for topic detection and extraction. Chapter 3 describes the proposed approach, including the tools and methodologies used. Chapter 4 shows the experiments carried out for extracting trending topics for a twitter user using document-pivot approach. Chapter 5 shows the experiments carried out for extracting trending topics for a twitter user using feature-pivot approach and applying the two approaches on different data sets. Finally, in chapter 6, we conclude our work.

Chapter 2. Approaches for topic detection and extraction

Topic detection and extraction can be done using supervised approaches as classification which requires a prior knowledge of the topics extracted or unsupervised approaches depending on clustering related items together without prior knowledge of the topics. In our research we chose to focus on the unsupervised techniques.

In this chapter we are presenting the two mostly used unsupervised approaches of topic detection and extraction which are the document-pivot and the feature-pivot approach.

In the document-pivot approach we are introducing different clustering techniques used for topic detection and different topic extraction approaches.

In the feature-pivot approach we are introducing how researchers used this approach for topic extraction from twitter and similar micro-blogging services.

Finally we are summarizing our findings that will guide us through finding the best approach for trending topic extraction for a Twitter user.

2.1. Document- pivot approach

In this approach tweets are clustered so each cluster represents a topic. Different clustering techniques have been used for this task. Various results were presented some of them will be mentioned in the literature. Results varied from a domain to another in some techniques. Actually clustering is considered the key role in this task, as the higher the quality data is clustered the higher the quality of results achieved in further tasks.

Clustering is an unsupervised technique that has no previous information about the data. For that validation metrics must be used to check how accurate the results are. Choosing the right clustering technique is considered a challenge in this task.

2.1.1. Clustering Approaches:

Data needs to be processed as a first step for clustering. Different presentation of data has been discussed in various researches.

General steps for pre-processing is presented by (Makkonen, 2009)

Pre-processing of data:

1. Identify individual words and reduce the typographical variation. (tokenization)
2. Remove non-informative words. (stop-words removal)
3. Reduce morphological variation. (stemming)
4. Compute the term-weights. (using TFIDF or other models)
5. Build the vector.

Clustering can be divided generally into hierarchical clustering and partitional clustering (Rui Xu & Wunch, 2009). In the following section we are going to present the most common used techniques related to our research.

Before proceeding in the discussion of various techniques we have to know how certain data will be in one cluster while others in different ones, that's what is called proximity measures. Simply proximity measures are measures of similarity between data. Similar data are grouped together into one cluster. Various measures are used, one of the most commonly used one which is used in most of the literature is the cosine similarity. We can return to the book by (Rui Xu & Wunch, 2009) which discusses in details various clustering techniques.

2.1.1.1. Hierarchical Clustering

In hierarchical clustering it starts grouping similar items bottom-up till reaching a single cluster which is called Agglomerative clustering, or top-down by dividing them into groups to maximize the objective function (Young & Sycara, 2004). Both methods results in a structure of data called dendogram. The root node represents the whole data set and each node represents a cluster. We can cut at any stage of the Dendogram to show the relation between clusters at certain stage.

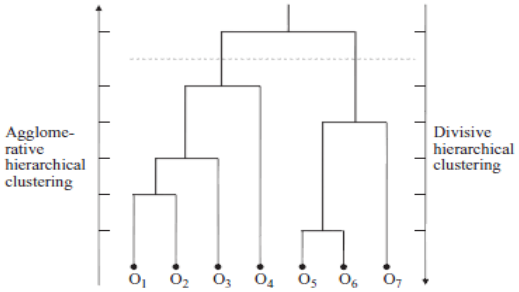

Figure 2-1 Dendogram, showing both techniques of hierarchical clustering. (Rui Xu & Wunch, 2009).

2.1.1.2. Agglomerative Hierarchical clustering

In this technique each point is represented as a cluster. Proximity matrix is calculated for each cluster to determine which pairs to be merged. This process continues till one cluster left. The merging of pairs of clustering depends on the minimal distance between them. Calculating this distance is done using various methods such as Single Link, Complete Link and Average Link. Those can be considered the most common techniques used. Figure (2-2) shows the algorithm for this technique.

(Dai & Sun, 2010) used agglomerative clustering with time decay to identify events in news. Time decay feature helps clustering stories about the same event. For example if we have two stories of a plane crash at a specific location, they may be talking about the same event reported by different sources or two stories about different events happened at different times but happened to be similar. Also it helps detect new events as an event is defined as a newly happened action. In their work they developed an approach to calculate the weights of different features. They used cosine similarity for calculating similarities between stories multiplied by the decay time factor.

(Dai et al 2010) improved the agglomerative hierarchical clustering algorithm based on the average link method. The improvement is achieved through splitting the original algorithm into two steps. The 1st step is calculating the similarity of each pair of two topics, and directly combining them if the similarity between them is higher than some threshold. Then the topic

model is rebuilt. The 2nd step is performing the universal agglomerative hierarchical clustering algorithm. The threshold is determined empirically. They also added more weight for feature terms occurring in the title of news story so its weight increases when calculating similarity.

(Young-dong et al, 2009) used hierarchical agglomerative clustering technique in their work. They used it to establish the hierarchical topic tree as the dendogram represents the same hierarchy of the general topic and sub-topics scheme.

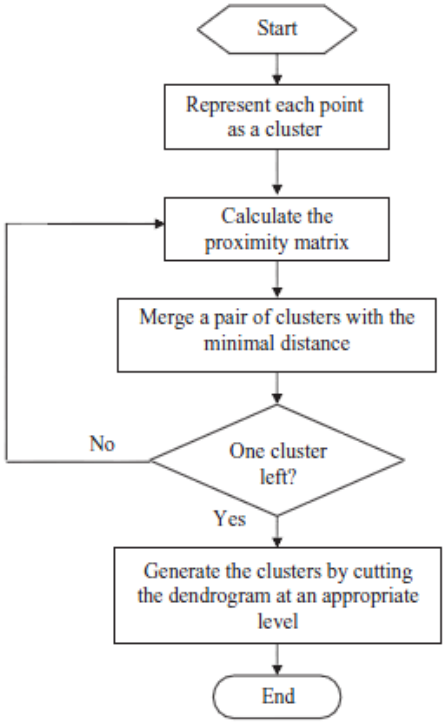

Figure 2-2 Flow chart showing the algorithm for the agglomerative hierarchical clustering (Rui Xu & Wunch, 2009)

(Huang & Cardenas, 2009) used hierarchical agglomerative method to group articles into clusters of same events. Their work aimed extracting hot events from news feeds.

Though clustering techniques is used to cluster related documents together some works tackled using clustering for topic extraction as well. (Okamoto & Kikuchi, 2009) used agglomerative clustering for topic extraction from blog entries within a neighborhood.

2.1.1.3. Divisive Hierarchical clustering

This technique works in the opposite way of the agglomerative way. The data set at the starts is in one single cluster then it's divided in successive operations till each node represents a cluster that can no more be divided. The figure below shows the algorithm for this technique using a famous heuristic approach called DIANA (divisive analysis) (Rui Xu & Wunch, 2009).

Hierarchical clustering still has its drawbacks, it lacks robustness and it's sensitive to noise (Rui Xu & Wunch, 2009). Once an object is assigned to a cluster it will not be considered again which leave no room for correcting an error happened during the beginning (Young & Sycara, 2004). Its computational complexity is at least $O(n^2)$ which is not suitable for dealing with very large data sets.

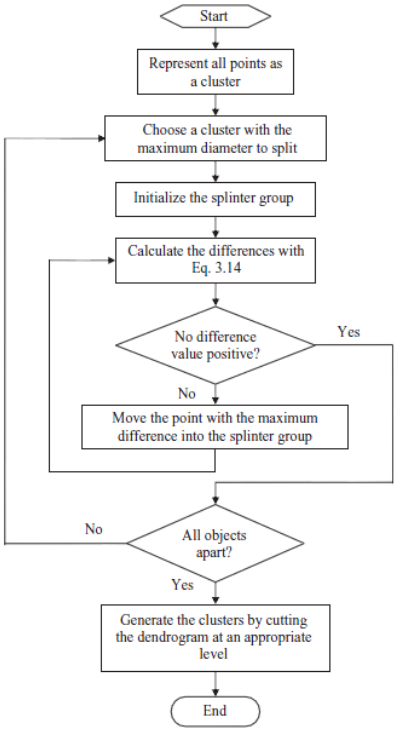

Figure 2-3 DIANA algorithm for divisive hierarchical clustering (Rui Xu & Wunch, 2009).

2.1.1.4. Partitional clustering

This technique assigns data into K clusters. It is based on optimizing a certain criterion. This criterion defines the homogeneity of the objects in the cluster. The sum of squared error criterion is defined as :

$$J_s(\Gamma, M) = \sum_{i=1}^{K} \sum_{j=1}^{N} \gamma_{ij} \left\| x_j - m_i \right\|^2$$

Where

$\Gamma = \{\gamma_{ij}\}$ is a partition matrix, $\gamma_{ij} = \begin{cases} 1 & if\ x_j \in cluster\ i \\ 0 & otherwise \end{cases}$ with $\sum_{i=1}^{K} \gamma_{ij} = 1\ \forall j$

$M = [m_1, \ldots, m_k]$ is the cluster prototype or centroid matrix

$m_i = \frac{1}{N_i} \sum_{j=1}^{N} \gamma_{ij} X_j$ is the sample mean for the i^{th} cluster with N_i objects

K is the number of clusters, N is the number of objects in a cluster

The partition that minimized the sum of squared error criterion is considered as optimal and is called the minimum variance partition (Rui Xu & Wunch, 2009).

K-means algorithm: It is one of the most known and used clustering algorithm. It minimizes the criterion of the sum of squared error using an iterative optimization procedure.

The algorithm of this technique goes as follows:

1. Initialize a K-partition randomly or based on prior knowledge. Calculate the cluster prototype matrix.
2. Assign each object in the data set to the nearest cluster C_i
3. Recalculate the cluster prototype matrix based on the current partition.
4. Repeat steps 2 and 3 until there is no change for each cluster.

K-mean algorithm was used by (Zhang et al, 2009) for topic detection.

Bisecting k-mean algorithm: is basically choosing two elements that have the largest distance as seeds for two clusters then proceed by assigning items to the nearest cluster to them from either seeds. (Wartena & Brussee, 2008) used the induced bisecting k-mean algorithm for their experiment in topic detection by clustering key words of documents. They also experimented with agglomerative hierarchical clustering, for their experiment the k-means algorithms performed better.

(Wang et al, 2008) discussed the use of incremental clustering for automatic topic detection. They proposed a new topic detection method called TPIC which adds the aging nature of topics to pre-cluster stories. Bayesian Information Criterion (BIC) is used to estimate the true number of topics. They compared their method to k-means and CMU and they achieved high performance by their proposed method.

2.1.2. Topic extraction Approaches

We just do not need to know that a set of tweets are related and belongs to a certain topic, but also we want to know the topic these tweets discuss. In this section we are going to discuss how topics can be extracted.

Witten et al, (1999) developed KEA which is a tool for key-phrase extraction. It identifies candidate key phrases using lexical methods, calculates feature values for each candidate, and uses a machine- learning algorithm to predict which candidates are good key phrases.

(Tomokiyo & Hurst 2003) used the statistical language model in their work. Their approach is to use point wise KL-divergence between multiple language models for scoring both phraseness and informativeness, which can be unified into a single score to rank extracted phrases. Phraseness is about how a set of words can be considered a phrase. This can differ based on user criteria. Informativeness is about how a phrase is informative about what the document is about.

(Jain & Pareek, 2009) used part of speech tagging in their work, formatting features and position of words in their work. Their results achieved high matching against the annotated data.

(Wang et al, 2008) used semantic information for automatic key phrase extraction in their work. Their method is divided into two stages. The first one is to select candidates, in this stage all phrases are extracted from the document, a word sense disambiguation method is used to get senses of phrases, case folding stemming and semantic relatedness between candidates is performed for term conflation. The second stage is called filtering stage, where four features are used to compute for each candidate, tf-idf, first occurrence of a phrase, length of a phrase, and coherence score which measures the semantic relatedness between the phrase and other candidates. They compared their results to KEA and achieved higher performance and showed their method is not domain-specific.

(Lopez et al, 2010) worked on automatic titling of electronic document with noun phrase extraction. It is based on the morpho-syntactic study of human written titles in a corpus of various texts. The method is developed in four stages: Corpus acquisition, candidate sentences determination for titling, noun phrase extraction in the candidate sentences, and finally, selecting a particular noun phrase to play the role of the text title. They call this approach ChTITRES approach.

(El-Beltagy & Rafea, 2008) developed a system called KP-Miner. It extracts key phrases from English and Arabic texts. This system has the advantage that it's configurable as the rules and heuristics adopted by the system are related to the general nature of documents and key phrase.

(Huang & Alfonse, 2009) in their work they relied on extracting hot events from news feeds. The cluster with more hot terms or with high weighted hot terms is examined for hot terms. Hot terms are mostly topical terms i.e. they express the topic title.

The study presented by (Xie et al, 2011) discussed the optimization design of subject indexing. Their work is based on the word frequency statistics. They took into consideration the word length, position and frequency in the weighting coefficient of the word. They considered long words as more specialized and short words are more generic.

2.2. Feature-pivot approach

This approach used recently in many researches for Twitter, since it fits the task of event detection better, where documents (tweets) are of short length. (El Sawy et al, 2014) presented a

news portal platform called TweetMogaz that generates news reports from social media content. They focus on Egyptian politics, Syrian conflict, and international sport. They use an adaptive information filtering technique for tracking tweets relevant to specific topics.

(Cataldi et al, 2010) tackled Twitter for extracting emerging topics. First, they extract the contents (set of terms) of the tweets and model the term life cycle according to a novel aging theory intended to mine the emerging ones. The term is emerging if it frequently occurs in the specified time interval and it was relatively rare in the past. For the content importance depending on the source, they analyze the social relationships in the network with the well-known page rank algorithm in order to determine the authority of the users. Finally, a topic graph is constructed connecting the emerging terms with other semantically related keywords, allowing the detection of the emerging topics, under user-specified time constraints Machine learning approach.

(Li et al, 2012) presented a system named Twevent, the system detects burst phrases based on frequencies then performs KNN clustering to produce disjoint clusters.

(Zhao et al, 2014) presented a system for topic detection and topic sentiment analysis on Twitter in China. They used hash tags as topics' titles, and then applied hierarchical clustering to cluster related topics together.

(Rosa et al, 2014) proposed a technique called Twitter Topic Fuzzy fingerprints. They compared their results with support vector machines (SVM) and k-nearest neighbors (kNN). Their technique outperforms the other two. They focused on data set of Portuguese language tweets and the respective top trends as indicated by Twitter.

(Aiello et al, 2013) compared six topic detection methods on three Twitter datasets related to major events. They proposed a novel method based on n-grams co-occurrence and df-idf topic ranking which performed better than the state of the art techniques.

(Parikh & Karlapalem, 2013) proposed an approach that detects events by exploring their textual and temporal components. Their results showed that they are able to detect events of relevance efficiently.

2.3. Summary

After reviewing the two approaches we found that the document-pivot approach was firstly used in topic detection from news streams and blogs before micro-blogging appear. It is relying mainly on clustering similar documents together and presents them as one topic. Many clustering techniques were used in this task. The main challenge in this task is to find the proper clustering technique that is efficient enough to detect the topics from the data. The major drawback of clustering that not all techniques can work with massive amount of data and some of them requires a prior knowledge of the number of clusters like in k-means clustering. To reach our objective of extracting the topic we need a further task under this approach called topic extraction. Some approaches based on statistical and linguistic approaches are used to achieve this task. For this task to work properly the documents in the clusters should be of high quality. By applying this approach on Twitter it is challenging as the size of the tweet does not exceed 140 characters which is way smaller in size than the documents used before. Also the structure of the tweet is way different than the structure of a document.

Recently many researchers adopt the feature-pivot approach which they found more suitable for short documents like tweets more than the document-pivot approach. In this approach the trending words are extracted as features in the first step then these features are grouped together representing the topic. The technique of grouping those features together is the main challenge of this approach. As finding words related to the same topic can be tricky in some domains.

Since the style of writing and dialect language of each region affects the nature of tweets in a great way we are focusing on the Egyptian user to match his/her interests.

In the light of those findings we are investigating the effect of both approaches on extracting trending topics for a twitter Egyptian user during a specific period of time.

Chapter 3. Proposed Approach

The outcome of the proposed methodology is building an unsupervised system for trending topic extraction for Arabic twitter user within a specific period of time. The sections of this chapter describe the steps needed to build such system. The first task is to crawl a development data set which is a sample of tweets to help for deciding on the algorithms and parameters that will be used by the document pivot and feature pivot approaches. The second task is to prepare the data by annotating the tweets manually with the appropriate topic(s) and preprocessing the crawled data automatically. The third task is to build a system based on document pivot approach. The fourth task is to build a system based on feature pivot approach. The fifth task is to validate the two approaches using data of different sizes from different domains.

3.1 Crawling data

First of all we needed to get data from Twitter. The Twitter platform offers access to data, via APIs. Twitter has two APIs. The Twitter REST API methods allow developers to access core Twitter data. This includes updating timelines, status data, and user information. It also includes the Search methods which allow developers to retrieve Twitter Search data. The Streaming API provides near real-time high-volume access to Tweets in sampled and filtered form. The Streaming API is distinct from the REST API as Streaming supports long-lived connections on a different architecture.

A Tweets' crawling tool was developed making use of the REST API v1.1. It returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users he/she follows. The home timeline is central to how most users interact with the Twitter service. The maximum number of tweets can be retrieved in a call is 200. The maximum number of calls in an hour is 4. (Twitter API documentation,2015)

With the increase of Arabic users on Twitter, it became a popular social media tool. The availability of Twitter on Mobile phones made it easier to use among lots of users. After the Arab Spring, Twitter became a main source of information about what is happening right now.

People started to check twitter the very first thing before any other media sources. The short nature of tweets made the news information brief and into the point which is more convenient to lots of people who wants to know what's happening without reading long articles.

As posting on Twitter usually done by normal users, they can post in any language they want. In the Arab world especially in Egypt, users tend to use dialect language more than standard Arabic language except for some news accounts that use it more frequently.

For the above reasons we needed to keep in mind the nature of Egyptian posts while analyzing the data.

Extracting most frequent hash-tags may seem a straight forward and simple approach, but applying it to Egyptian tweets was different. In our preliminary experiments we faced some problems like:

1. Hash-tags misuse:
    - Most of news accounts include their names as hash-tags in the text of the tweet which bias the clustering process.

      Example:

ابراهيم الدميري وزير النقل افتتاح المرحلة الثانية من الخط الثالث لمترو الانفاق خلال شهر أبريل 2014 #سى بى سى Egypt
البنك المركزى يقرر مد فترة العمل بمبادرة دعم قطاع السياحة حتى ديسمبر 2014 #سى بى سى #Egypt
جلال سعيد محافظ القاهرة حملات مكثفة لتجديد شوارع منطقة جاردن سيتى ومهلة لرئيس حي غرب خلال 48#egypt ساعة لأعمال النظافة #سى بى سى
عبد الرحيم مصطفي المتحدث باسم هيئة موانئ البحر الأحمر ميناء الزيتات أستقبلت 8500 طن بوتاجاز سائل قادمة من ميناء ينبع السعودي #سى بى سى

    - Using lots of hash-tags in the tweet makes it difficult to put it under the proper topic group.

      Example:

      المعادي الإعلام#تطالب البحرين#و الإمارات#و السعودية#بعدم دعم قطر# ..آخر خبر

- Using hash-tags in a very general way that doesn't relate directly to the content of the tweet.

  Example:

  أثناء زرع عبوة ناسفة على الحدود "السورية - #حزب_الله الجيش الإسرائيلي يعلن إصابة عنصرين من الإسرائيلية" #Egypt #Syria

  وسقوط العديد من القتلى والجرحى..ومصدر يؤكد:سيارات مفخخة #بغداد سلسة تفجيرات متزامنة تضرب #الشيعة استهدفت أماكن تسكنها غالبية من #Iraq #Egypt

2. Misuse of trending hash-tags:

   Users in Egypt tend to use meaningless hash-tags to hit the top 10 trending hash-tags.

   Example:

   #أجمل_لحظات_حياتي_لما

   #عيبنا_اننا

   #اللي_جاتلهم_رسائل_غريبه_بيقولو_بعض

   #انا_ماعنديش_مانع

We found that depending only on hash-tags won't achieve our objective so we are investigating different approaches to find the efficient way to extract trending topics for a Twitter user in Egypt.

3.2 Annotating and preprocessing data

In order to evaluate our results we need to have an annotated data to compare the results to. Data sets are annotated by giving each tweet a topic. The following sections explain the process of annotating data for the baseline and different data sets used for validation. It also includes the number of annotated trending topics and the number of tweets in each data set.

3.2.1. Annotating the baseline

As the annotating process is very time consuming we made a call every hour on October 2nd 2014 from 12:00 pm to 11:30 pm. The tweets are crawled from news domain during the celebration of the feast and the pilgrim. Tweets are annotated so every tweet belongs to a topic. Topics contain less than 5 tweets are removed from the dataset. Topics contains more than 20 tweets are considered trending topics. Table (3-1) contains data statistics.

The results of the extracted trending topics from the developed systems will be compared manually to the annotated trending topics to calculate the recall and F1 measure values.

Table 3-1 Baseline data statistics

Number of tweets	Number of Trending topics
1266	18

3.2.2. Annotating different data sets

To validate the results of applying the document pivot and the feature pivot approaches on different data sets, we collected several data sets of sizes 200, 400, 600, and 1200 tweets from three different domains; sports, entertainments, news.

Those sets of data have been annotated with the help of human participants according to the recommendation and approval of the Institutional Review Board (IRB) for CASE #2014-2015-155 .

The annotation guidelines used are as follows:

1. Define the topic of the tweet it is related to. Maximum three words are used to define the topic.

2. If the category is tricky or the tweet could be related to more than one topic, three people should agree to the closest topic. If it is still hard to decide the topic a voting between the participants must be held.
3. If a participant has other opinion about an annotation of a tweet s/he can explain his point of view to the other participants, if three of them agreed with him/her the annotation could be changed otherwise it couldn't.
4. Every user will be assigned 600 tweets to annotate.
5. We will rely on the participant's sole judgment on his/her assigned annotated tweets.

Principles to keep in mind when annotating

1. Tweet event: a good understanding of the tweets sentences.

2. What: what happened during the event.

3. Who: who (person, organization) was involved in the event, who wrote the tweet.

4. When: when the event occurred.

5. Where: where the event occurred.

Table (3-2) shows the statistics of the different data sets collected and annotated.

The sports data sets were collected on 1st of November 2015 during the matches of the Egyptian league between 5:00 pm and 7:30 pm with a call every half an hour results in 200 tweets per call.

The entertainment data sets were collected on 30th of June 2015 during Ramadan between 8:00 pm and 10:30 pm with a call every half an hour results in 200 tweets per call.

The news data sets were collected on 6th of October 2015 during the celebration of the 6th of October victory between 1:00 pm and 3:30 pm with a call every half an hour results in 200 tweets per call.

The results of the extracted trending topics from the developed systems will be compared manually to the annotated trending topics to calculate the recall, precision and F1 measure values.

Table 3-2 Statistics of different data sets

Domain	Number of Tweets	Number of Trending Topics
Sports	200	2
Sports	400	3
Sports	600	4
Sports	1200	5
Entertainment	200	2
Entertainment	400	3
Entertainment	600	6
Entertainment	1200	8
News	200	1
News	400	2
News	600	5
News	1200	10

3.2.3. Preprocessing

After the tweets being crawled they need to be preprocessed so they can be analyzed. The preprocessing phase consists of:

1. Removing urls and punctuation marks except the '_' symbol that is used in hash-tags so the tweet text is kept the same.
2. Removing account names:

   To handle the problem of including account names of most of the news accounts into the tweet's text, we could extract the screen name of the user account. Then if it's mentioned in the tweet's text it's removed from the tweet during the preprocessing phase.

3. Stop Words Removal:

Due to the lack of a stop words list for the Egyptian Dialect, and due to the nature of Egyptian tweets, some words occur very frequently and meaningless, we needed to build our own list. Although there is an existing list of 128 words presented by (Shoukry Amira, 2013) it was not comprehensive enough so we decided to increase these stop words from the data collected.

In this phase a call made every half an hour to build a corpus of 9458 tweets collected on Oct 2nd 2014 from 12:00 am till 11:30 pm. This corpus will be used to identify stop words list. Unigrams are extracted, and their frequencies are identified. We divided the frequency ranges into three ranges: from 0 to 10 times, from 10 to 100 times, and from 100 to 1000 times.

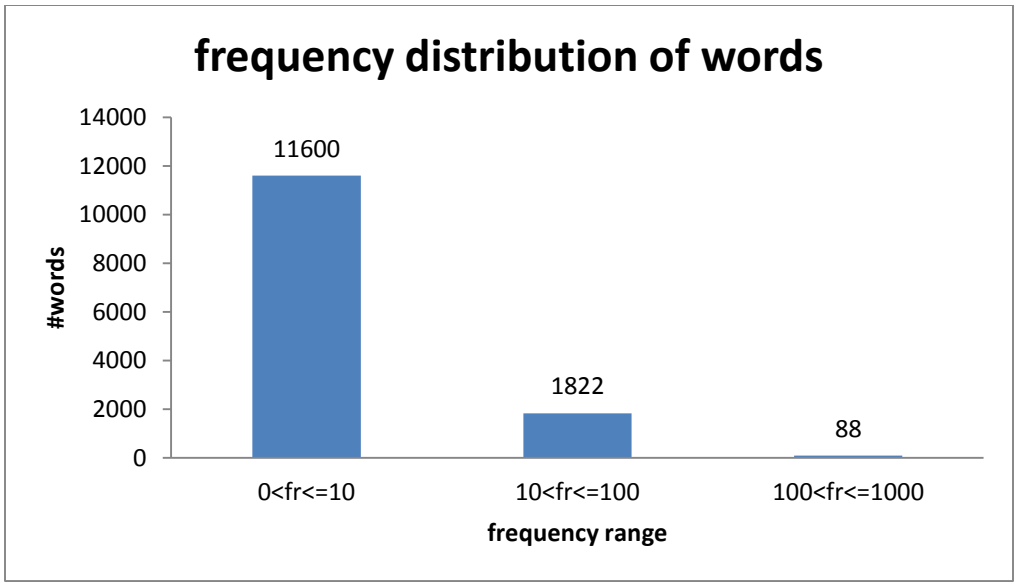

Figure 3-1 Frequency distribution of unigrams of the corpus

From 13510 unique words, those with frequency range between 10 and 1000 are filtered manually to produce a list of stop words. Some words need to be kept although they occur frequently like "الثورة" ,"مصر" and consequently the stop words are examined manually. A stop words list of 150 words was produced; where 22 new words were added to the existing list mentioned earlier.

3.3 Developing a Topic Extraction system based on document pivot approach

In this section we are investigating the impact of the applying the document pivot approach on the baseline data.

First we are introducing the steps for building a baseline based on the document pivot approach, and then we are investigating the impact of different clustering techniques, the feature representation and the different topic extraction methods on the extraction of trending topics for a twitter user.

3.3.1. Develop a Baseline System

In this section we are developing baseline system for tweets collected from a user timeline over 10 hours on 2nd of October 2014 from news domain during the celebration of the feast and the pilgrim. The tweets are represented using tf-itf vector space model, clustered using hierarchical agglomerative technique, then the most frequent hash-tags from each cluster are extracted to be topic title candidates

a. Vector representation

Vector space model is built using tf-itf for each word in a tweet, where tf is term frequency in the tweet and itf is the inverse tweet frequency in all tweets.

$itf = \log \frac{N}{n_i}$ Where N is the total number of tweets, n_i is number of tweets containing the term.

b. Clustering

We used a tool called Cluto 2.0 for clustering; hierarchical agglomerative clustering technique is used for clustering tweets together in the baseline.

The tool requires the number of resulting clusters as an input ahead of the clustering process. We are investigating different values of K range from 10 to 300 and record the performance at each value.

The results are compared to the annotated data to identify the value of k at which we could achieve the highest recall and F1 measure.

c. Topic extraction method

For each cluster the most frequent hash-tags are extracted to represent the topic of the cluster.

Each hash-tag extracted is compared against the account name of the author of the tweet, if they match, the hash-tag is not considered to overcome the misuse of hash-tags by the news accounts.

3.3.2 Investigate the impact of different clustering techniques

In order to investigate the impact of different clustering techniques on the results of extracting trending topic for twitter user we are performing the following experiments:

a) Run k-means clustering with different values of k values ranges from 10 to 300 and compare the results to the baseline.

b) Run repeated bisecting k-means and validate the results against the baseline.
In this method, the desired *k*-way clustering solution is computed by performing a sequence of $k-1$ ***repeated bisections***. In this approach, the matrix is first clustered into two groups, and then one of these groups is selected and bisected further. This process continuous until the desired number of clusters is found. During each step, the cluster is bisected so that the resulting 2-way clustering solution optimizes a particular clustering criterion function, which is maximizing $\sum_{i=1}^{k}\sqrt{\sum_{v,u\in S_i} sim(v,u)}$ Where k is the total number of clusters, *Si* is the set of objects assigned to the *i*th cluster, v and *u* represent two objects, and sim*(v, u)* is the similarity between two objects. The similarity is calculated using different techniques determined by the user like cosine similarity and Euclidian distance. (Cluto 2.1, 2003)

c) Run biased agglomerative clustering with k values range from 10 to 300 and compare the results to the baseline.

In this method, the desired *k*-way clustering solution is computed in a fashion similar to the *agglomerative* method; however, the agglomeration process is biased by a partitional clustering solution that is initially computed on the dataset. When *biased agglomerative* is used, first a $\sqrt{n}$ way clustering solution is computed using the *repeated bisecting* method, where *n* is the number of objects to be clustered. Then, it augments the original feature space by adding $\sqrt{n}$ new dimensions, one for each cluster. Each object is then assigned a value to the dimension corresponding to its own cluster, and this value is proportional to the similarity between that object and its cluster-centroid. Now, given this augmented representation, the overall clustering solution is obtained by using the traditional agglomerative paradigm. (Cluto 2.1 ,2003)

The best clustering technique is selected and replace the clustering technique in the baseline. Topic extraction method is applied on the selected clustering technique solution. The results are evaluated against the annotated tweets and compared to the results of the baseline.

3.3.3 Investigate the impact of feature representation

In order to investigate the impact of feature representation we are doing the following:

a. Represent tweets using N-grams instead of tf-itf, cluster them with the chosen technique from the previous experiments with the k value identified. Topic extraction method using hashtags is applied and then results are evaluated against the annotated data and compared to the results of the baseline.
b. Represent them using a hybrid of N-grams and tf-itf (N-grams-itf) where each n-gram is represented by its frequency in the tweet multiplied by its inverse frequency in the whole tweets.

   To determine the n-grams used as features, the frequency distribution of n-grams is calculated so n-grams that occur more than 10 times is included in the features list.

   i. Identifying unigrams:

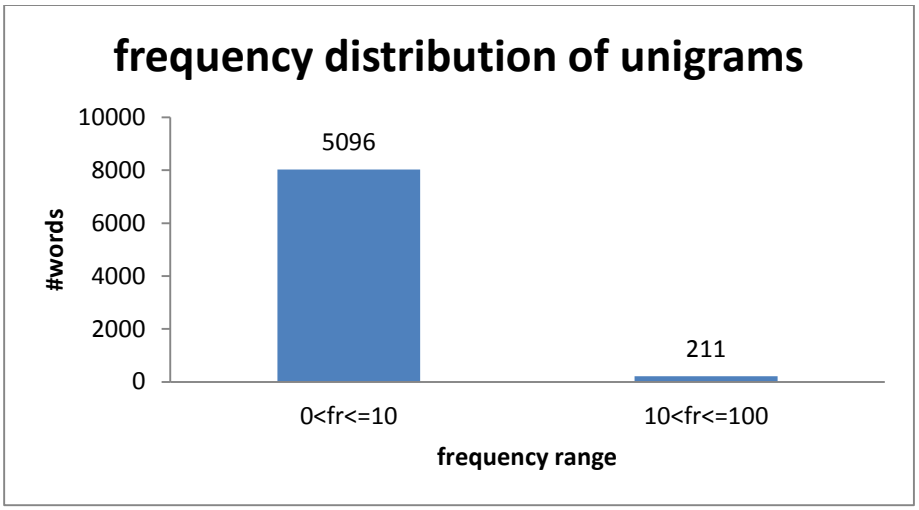

Figure 3-2 Frequency distribution of unigrams of tweets

From the above figure we can find that 211 unigrams is included in the features list.

ii. Identifying bigrams:

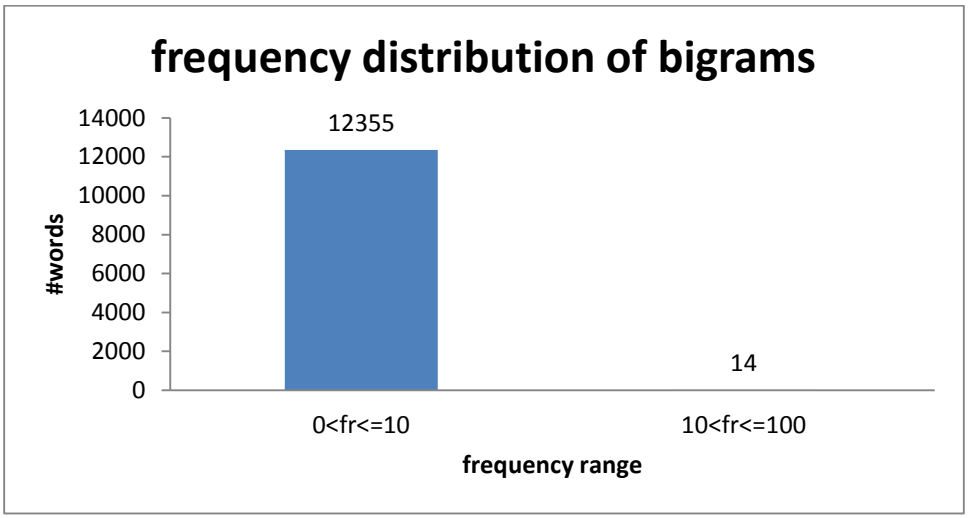

Figure 3-3 Frequency distribution of bigrams of tweets

From the above figure we can find that 14 bigrams is included in the features list.

iii. Identifying trigrams:

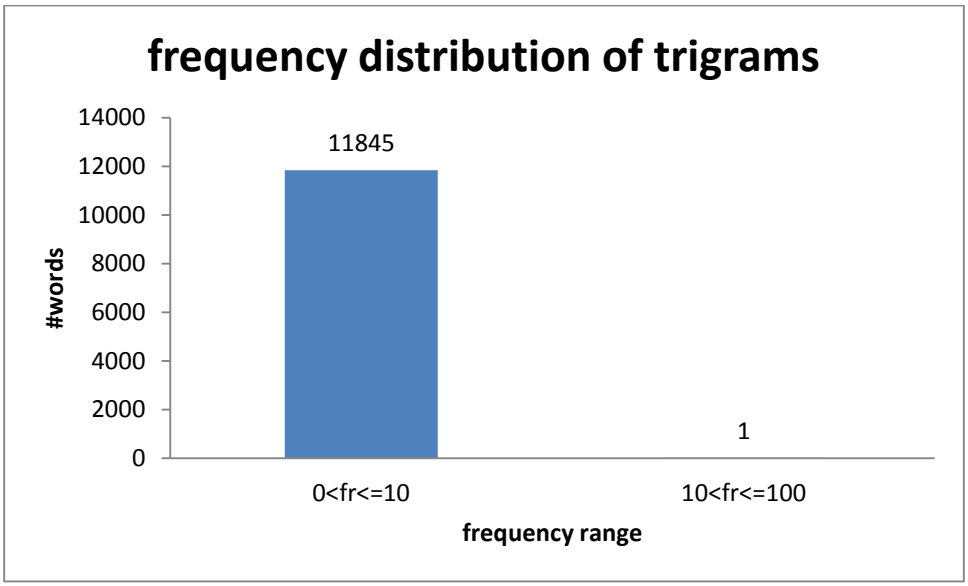

Figure 3-4 Frequency distribution of trigrams of tweets

From the above figure we can find that only one trigram is included in the features list. After representing the tweets using this method they are clustered using the chosen clustering technique and the identified k value from previous experiments. The topic extraction method using hashtags is applied then the results are evaluated against the annotated tweets and compared to the results of the baseline.

3.3.4 Investigating different topic extraction methods

In order to investigate the impact of different topic extraction methods, we are performing the following:

1. Extract most frequent bigrams from each cluster to represent the topic, and validate the results against the baseline.
2. Extract most frequent unigrams not included in any bigrams alongside with most frequent bigrams from each cluster, and validate the results against the baseline.
3. Extract most frequent trigrams alongside with unigrams and bigrams not included in any trigram, and validate the results against the baseline.
4. Determine the best combination of extracted n-grams.
5. Evaluate the results against the topic extraction method in the baseline.

3.4 Developing a Topic Extraction System based on Feature Pivot Approach

Methods of this approach are closely related to topic models in natural language processing, namely statistical models to extract sets of terms that are representative of the topics occurring in a corpus of documents. The common framework that underlies most approaches in this category first identifies trending terms (keywords) and then group them together based on their co-occurrence in the documents so they represent the topic label. (Luca et al, 2013)

Clustering those keywords is based on what is called content similarity, where keywords of the same topic appear together in tweets about that topic.

Keywords can be unigrams, bigrams, or trigrams; in our work we focus on using unigrams as we found from our observations that in Egypt a lot of events are described in only one word like: "العيد" and "الحج"

To identify those keywords, cluster them together, and represent the trending topic we implemented the following algorithm.

The algorithm goes as follows:

1. The set of tweets collected over a specific time period is preprocessed by removing stop words, punctuation marks, and account names of the author of the tweet if it appears in the tweet.
2. The set of tweets is tokenized (words are separated) and all unigrams are extracted.
3. Based on the Frequency Distribution of Unigrams, figure (3-2) showed that the meaningful unigrams usually have a frequency between 10 and 100, so we filtered the unigrams to only select those that occur more than 10 times in the set of tweets.
4. From that set of unigrams, get unigrams with frequency more than or equal to the average frequency (θ_1) of the set resulting from step 3 (formula.1), these unigrams are put in a set called the significant unigrams.

$$Avg.Freq = \frac{\sum_{x=1}^{n} Freq(h_x)}{n} \qquad (1)$$

Where $Freq(h_x)$ is the frequency of unigram h_x and n is the number of unigrams occurred more than 10 times in the set of tweets.

5. For each significant unigram, get the set of associated tweets where this unigram occurs.
6. From each set of associated tweets, the unigrams of these tweets are extracted so their proportional frequency (PF) (formula 2) is more than or equal to the average proportional frequencies (θ_2) of the unigrams in this set of tweets (formula 3). This set of unigrams is called the frequent common unigrams (FCU).

$$PF(u_s) = \frac{Freq(u_s)}{\sum_{s=1}^{z} Freq(u_s)} \qquad (2)$$

$$Avg.PF = \frac{\sum_{s=1}^{m} PF(u_s)}{m} \qquad (3)$$

Where $PF(u_s)$ is the proportional frequency of the unigram u_s extracted from the set of tweets, PF is the average proportional frequency of the unigrams extracted from the set of associated tweets, z is the number of unigrams in a set of tweets. (Parikh & Karlapalem, 2013)

Proportional frequency is used in this step to extract the frequent common unigrams (FCU) from the associated sets of tweets. As those sets contains relatively small number of tweets in contrast with the whole data set.

7. From 5 &6, we can see that for every significant unigram, there is an associated set of tweets, and a set of associated frequent common unigrams (FCU).
8. To cluster the significant unigrams (keywords) representing the trending topics, we check for content similarity between the tweets where those significant unigrams occur.
9. Checking for content similarity is done on two levels:
    a. Level 1: Get ordered pairs of significant unigrams (S_i, S_j) that their number of common associated FCU of S_i and S_j exceeds a certain threshold (θ_3). The threshold is a percentage of the number of associated FCU of both significant unigrams.
    b. Level 2: For each pair of significant unigrams (S_i, S_j) search for all pairs that have S_j as the first significant unigram (S_j, S_k) such that number of common associated FCU of S_j and S_k exceeds a certain threshold (θ_4) and combine them into a triple item (S_i, S_j, S_k). The threshold is a percentage of the number of associated FCU of both significant unigrams.
    c. Associated tweets of S_i, S_j and S_k are combined together in a way that no tweet is replicated.
    d. If the number of combined tweets exceeds the trending threshold (α) which is set to 20 tweets then this topic is trending.
    e. The significant unigrams (keywords) grouped together representing the topic. The tweets of the topic are also presented.

Figure (3-5) shows the feature pivot algorithm.

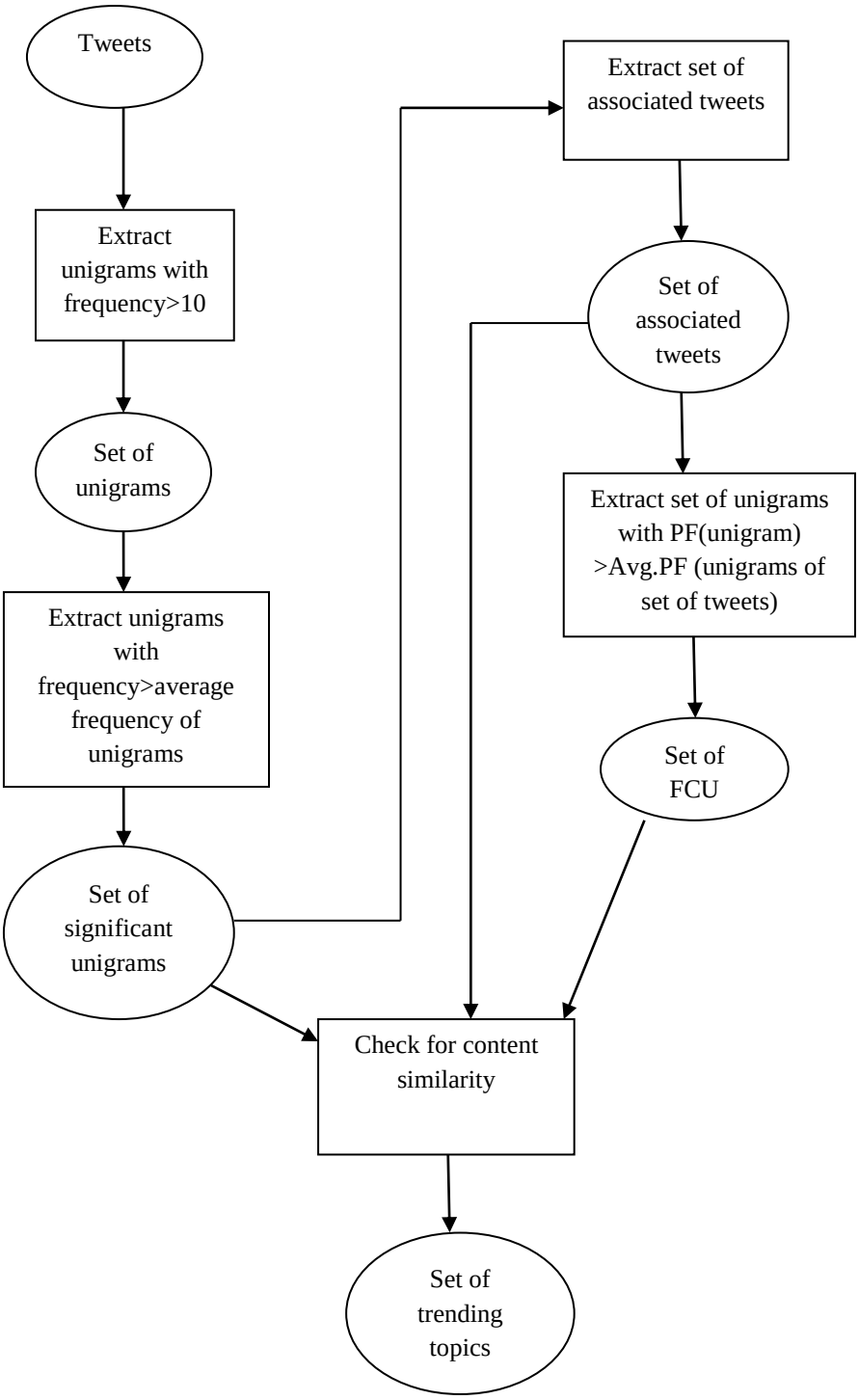

Figure 3-5 Feature Pivot algorithm

In order to investigate the effect of applying the feature-pivot approach on the tweets of the baseline data set the following experiments are being implemented:

I. Investigate the effect of different values of the threshold of the first level of content similarity (θ_3) on the extraction of trending topics for a twitter user. This is done by using the tweets of the baseline data set. The feature pivot algorithm is applied by setting the threshold (θ_3) to different values and fixing the value of the threshold of the second level of content similarity (θ_4) to an arbitrary value. The results are then evaluated against the annotated data to identify the value of (θ_3)

II. Investigate the effect of different values of the threshold of the second level of content similarity (θ_4) on the extraction of trending topics for a twitter user. This is done by using the tweets of the baseline data set. The feature pivot algorithm is applied by setting the threshold (θ_3) to the value identified from the previous experiments and set the value of (θ_4) to different values. The results are then evaluated against the annotated data to identify the value of (θ_4)

III. The results obtained by setting the thresholds of the first and second level of content similarity to the values identified from the previous experiments are compared to the results obtained by applying the document pivot approach to the baseline data set.

The pseudo code of the implementation is presented in the following algorithms. The implementation of these algorithms in Python can be found in appendix [B]

Algorithm 1 Trend_Topic_Extraction (*Tweets*)

---

list_of_unigrams = **extract_unigrams** (*Tweets*) *//extracting unigrams of all tweets in the data set*

θ_1 = **average_freq** (*list_of_unigrams*)

significant_unigrams = **extract_significant_unigrams** (*list_of_unigrams*, θ_1)

m=len(*significant_unigrams*)

//extracting associated tweets and associated frequent common unigrams for each significant unigrams

for i in range (1,m) **:**

associated_tweets_set[i] .append (**extract_tweets** (Tweets, *significant _unigrams[i]*))

associated_tweets_unigrams[i] .append(**extract_unigrams** (*associated_tweets_set[i]*))

θ_2 = **average_PF** (*associated_tweets_unigrams[i]*)

FCU[i] .append(**extract_FCU**(*associated_tweets_unigrams[i]* , θ_2))

end for

Content_similarity (*significant_unigrams*, associated_*tweets_set*, *FCU*, θ_3, θ_4 , α)

Algorithm 2 Content_similarity (*significant_unigrams*, associated_*tweets_set*, *FCU*, θ_3, θ_4 , α)

---

keywords= { a } //set of significant unigrams representing trending topics, initially contains an arbitrary value

t= 1 // index of number of trending topics

for i in range (1, len(*significant_unigrams*)) **:**

topic = [ ]

topic_tweets = [ ]

if (*significant_unigrams [i]* **not in** *keywords*) **:**

topic.append**(** *significant_unigrams[i])*

keywords.append(significant_unigrams[i])

Add_tweet_to_topic*(associated_tweets_set[i],topic_tweets)*

for j in range (i+1 , len (*significant_unigrams*)) **:**

if (**similar** (*FCU[i]*, *FCU[j]* , θ_3)**:**

topic.append**(** *significant_unigrams[j])*

keywords.append(significant_unigrams[j])

Add_tweet_to_topic*(associated_tweets_set[j],topic_tweets)*

```
                    for k in range (j+1 , len (significant_unigrams )) :
                        if (similar ( FCU[j] , FCU[k] , θ4 ):
                            topic.append( significant_unigrams[k])
                            keywords.append(significant_unigrams[k])
                            Add_tweet_to_topic(associated_tweets_set[k],topi
                            c_tweets)

                        end if
                    end for
                end if
            end for
        end if
        if  ( len ( topic_tweets[t] >= α ) :
            print "topic"+" "+t
            print topic
            print topic_tweets
            t=t+1
        end if
end for
```

Algorithm 3 Add_tweet_to_topic (*associated_tweets_set,topic_tweets)*

```
for tweet in associated_tweets_set :
    topic_tweets.append(tweet)
```

Algorithm 4 similar (*FCU1* , *FCU2* , threshold *)*

```
common = [ ]
flag = FALSE
for word1 in FCU1 :
        for word2 in FCU2 :
                if (word1 == word2 ):
                        common.append (word1)
                end if
        end for
end for
if ( len(common) >= (len (FCU1) + len (FCU2)) * threshold ) :
        flag = TRUE
end if
return flag
```

3.5 Validating the Systems Built Using Document Pivot and Feature Pivot Approaches

To investigate the effect of applying both approaches on different data sets, we collected several data sets of different sizes; 200,400, 600, and 1200 tweets, from three different domains; sports, entertainments, and news.

Those data sets were annotated with the help of human participants as mentioned in section 3.2.2

In order to validate our results the following is performed:

1. All data sets are annotated and preprocessed.
2. Document-pivot approach is applied to each data set separately by running the clustering algorithm proved to be the best from previous experiments, and topic extraction method investigated in the experiments.
3. Feature-pivot approach is applied to each data set separately using thresholds determined through experiments on the baseline data.
4. Validate the results against the manual annotation.
5. Apply Two-sample paired significance t-test on the achieved results to find out if the results of one of the approaches are significantly different than the other.

3.5.1 Evaluation

In order to evaluate our system, the results obtained are compared manually against the annotated data to build a confusion matrix to get the recall, precision and F1 measure values.

The Two-sample paired t-test is carried to find out if applying one of the approaches yields in significant better results or not.

I. Confusion Matrix

To evaluate the results of experiments, the number of extracted trending topics is recorded, and then a confusion matrix is built as follows:

- True positive (TP) when extracted topic matches the annotated topic.
- False positive (FP) when the extracted topic identify a topic as trending while the topic is not.
- False negative (FN) when annotation identify a topic as trending but the extraction method didn't.
- True negative (TN) when both the extraction method and the annotation didn't identify a topic as a trending topic.

Sample confusion matrix:

		Extracted Topics	
		True	False
Annotated topics	True	True positive instances	False negative instances
	False	False positive instances	True negative instances

Precision, Recall and F1 measure are used to evaluate the results.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

$$Recall = \frac{True\ positive}{True\ positive + False\ Negative}$$

$$F1\ measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

II. Two-sample paired significance t-test:

The two-sample paired significance test is a type of the student t-test used when we have two measures on the same subjects. For example if we want to compare the size of tumor before and after treatment for the same group of patients. (Zimmerman, 1997).

In our work we are applying the document pivot approach on different sets of data, and record the recall, precision and F1 measure of the results resulted from evaluating the results against the annotated data. Then we apply the feature pivot approach and record the same evaluation measure.

Afterwards we apply the t-test to measure how significant is the difference between the results achieved from applying the feature pivot approach and the document pivot approach.

There are two types of test: one-tailed and two-tailed. The choice of which test is to be used rely on the knowledge we have beforehand. (Kock, 2015) For example if our hypothesis is that there is an increase in performance related with applying an approach then we need a one-tailed test. As we need to test if there is a significant increase or not. On the other hand if our hypothesis is that there is a change in performance related with applying an approach then we need a two-tailed test. As we need to test if there is a significant increase or a decrease.

In our work our hypothesis will be that one of the approaches yields better results than the other. We are performing the test to accept or reject this hypothesis. So we will perform a one-tailed test as we need to test the significance of change in one direction only.

We have two values of significance in the test, the significance level $\alpha = 0.05$ which is the probability to accept our hypothesis. And the *p-value*, which is the probability of obtaining at least as extreme results given that our hypothesis is false. (Schlotzhauer,2007) If the *p-value* $< \alpha$ then there is a significant difference between the two groups of data.

Using the degree of freedom and the value of $\alpha = 0.05$ and a confidence interval of 90% we get $t_{critical}$ from the one-tailed t-test table at (Renee & James, 2011)

The following steps are used to perform the test:

Step 1: Calculate the mean values of each set of data, sum of difference between pairs, sum of square differences between pairs, and the standard deviation of the differences between pairs.

$$\overline{D} = \frac{\sum_{i=1}^{n} D_i}{n}$$

Where: $\overline{D}$ is the mean of differences between pairs, D is the difference between two pairs, and n is the number of pairs.

$$S_D = \sqrt{\frac{\sum_{i=1}^{n} D_i^2 - \frac{(\sum_{i=1}^{n} D)^2}{n}}{n-1}}$$

Where S_D is the standard deviation of the difference between pairs.

Step 2: Calculate $t_{obtained}$

$$t_{obtained} = \frac{\overline{D}}{\frac{S_D}{\sqrt{n}}}$$

Step 3: Calculate the degree of freedom = $n - 1$

Step 4: Extract $t_{critical}$ from the t-test table using the value of the degree of freedom at α =0.05, extract p-value for the p-value table found in (Piegorsch et al, 2005)

Step 5: Compare the $t_{obtained}$ and $t_{critical}$ and the *p-value* to α to prove or reject the hypothesis. The hypothesis is accepted when $t_{obtained}$ is greater than $t_{critical}$

Chapter 4. Trending Topic Extraction using Document-Pivot Approach

In this chapter we first present the baseline system that will be used to identify the clustering technique, the tweets features representation, and topic extraction method to develop the best trending topic extraction system that we can get using document-pivot approach. Different clustering techniques investigated, different tweets' features representations examined, and different methods for extracting topic from clustered tweets are described in sections two, three and four respectively.

4.1. Building baseline

4.1.1. Objective

The objective of this experiment is to build a baseline so further results are compared against it.

4.1.2. Method

To achieve our objective, the following is performed:

- Tweets are crawled and manually annotated as described in the methodology chapter 3
- Tweets are represented using tf-itf representation
- Hierarchical agglomerative clustering (agglo), using different k values range from 10 to 300 to determine the best k.is used
- The topic of the cluster is determined by the annotated tweets belonging to the same topic and occupies more than 50% of the cluster size.
- Consider hash-tags extracted from each cluster as the trending topics. Hash-tags are extracted from each cluster as follows:
    - Hash-tags occur more than or equal to 50% of the cluster size are extracted, the results are evaluated against the annotated topics.
    - Hash-tags occur more than or equal to 30% of the cluster size are extracted, the results are evaluated against the annotated topics.
    - Hash-tags occur more than or equal to 25% of the cluster size are extracted, the results are evaluated against the annotated topics.

4.1.3. Results

4.1.3.1. Clustering results

We performed 30 experiments for different values of K (numbers of resulting clusters) in the range between 10 and 300. Average Purity, Average Entropy, Average Intra-similarity and F1 measure were recorded as well as number of detected trending topics and their recall values.

Purity of a cluster is a measure of how the objects in a cluster are related to the same topic, the higher the better. Entropy is the measure of how the various classes of documents are distributed within each cluster. (Zhao & Karypis, 2001)

Given a particular cluster S_r of size n_r , the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{q}\sum_{i=1}^{q}\frac{n_r^i}{n_r}\log\frac{n_r^i}{n_r}$$

Where q is the number of classes in the dataset, and n_r^i is the number of documents of the i th class that were assigned to the r th cluster. The entropy of the entire clustering solution is then defined to be the sum of the individual cluster entropies weighted according to the cluster size. That is

$$Entropy = \sum_{r=1}^{k}\frac{n_r}{n}E(S_r)$$

Where k is the total number of clusters, n is the total sizes of all clusters.

The purity of a cluster is defined as:

$$P(S_r) = \frac{1}{n_r}\max_i\left(n_r^i\right)$$

The above formula represents the fraction of the cluster size that the largest class of documents occupies. The purity of the entire clustering solution is as follows:

$$purity = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r)$$

The number of detected trending topics is the number of trending topics detected by the clustering process; it is done by manually examining the clusters of high purity values that means the major number of tweets in them related to the same topic. If the tweets belong to a trending topic according to the annotated data then a trending topic is detected.

The value of purity and entropy are determined by feeding the tool CLUTO the annotation of each tweet, so it can calculate their values according to the tweets belonging to the same topic in each cluster. The total entropy and entropy of the clustering solution is the average of the purity and entropy of all clusters in the solutions. (Cluto 2.1, 2003)

Table 4-1 shows the results of clustering solutions at different values of k between 10 and 300.

Figure 4-1 shows the F1 measure of the detected trending topics, and figure 4-2 shows the recall value of the detected trending topics.

Table 4-1 Results of clustering using different values of k in range between 10 and 300

K (number of clusters)	Average Intra similarity	Purity	Entropy	F1measure	No. of detected trending topics	Recall
10	0.12475	0.368	0.538	0.347826	4	0.2222
20	0.11523	0.514	0.391	0.482759	7	0.38888
30	0.17281	0.584	0.333	0.555556	10	0.55555
40	0.21972	0.622	0.291	0.571429	12	0.66666
50	0.24046	0.659	0.25	0.595745	14	0.77777

60	0.24791	0.722	0.214	**0.62963**	**17**	**0.94444**
70	0.26565	0.743	0.196	0.596491	17	0.94444
80	0.28626	0.754	0.182	**0.610169**	**18**	**1**
90	0.29824	0.759	0.172	0.6	18	1
100	0.30849	0.761	0.165	0.6	18	1
110	0.32636	0.786	0.149	0.580645	18	1
120	0.34557	0.79	0.143	0.571429	18	1
130	0.35411	0.794	0.139	0.571429	18	1
140	0.36696	0.802	0.133	0.553846	18	1
150	0.37416	0.806	0.127	0.553846	18	1
160	0.38401	0.81	0.122	0.553846	18	1
170	0.38775	0.813	0.118	0.553846	18	1
180	0.39209	0.819	0.113	0.553846	18	1
190	0.40256	0.823	0.108	0.553846	18	1
200	0.40195	0.829	0.103	0.553846	18	1
210	0.41853	0.831	0.1	0.545455	18	1
220	0.42781	0.835	0.096	0.545455	18	1
230	0.43421	0.838	0.091	0.545455	18	1

240	0.44122	0.846	0.086	0.537313	18	1
250	0.44903	0.85	0.083	0.537313	18	1
260	0.45264	0.854	0.08	0.537313	18	1
270	0.45874	0.857	0.077	0.537313	18	1
280	0.46428	0.859	0.075	0.537313	18	1
290	0.47014	0.86	0.073	0.537313	18	1
300	0.47604	0.86	0.072	0.537313	18	1

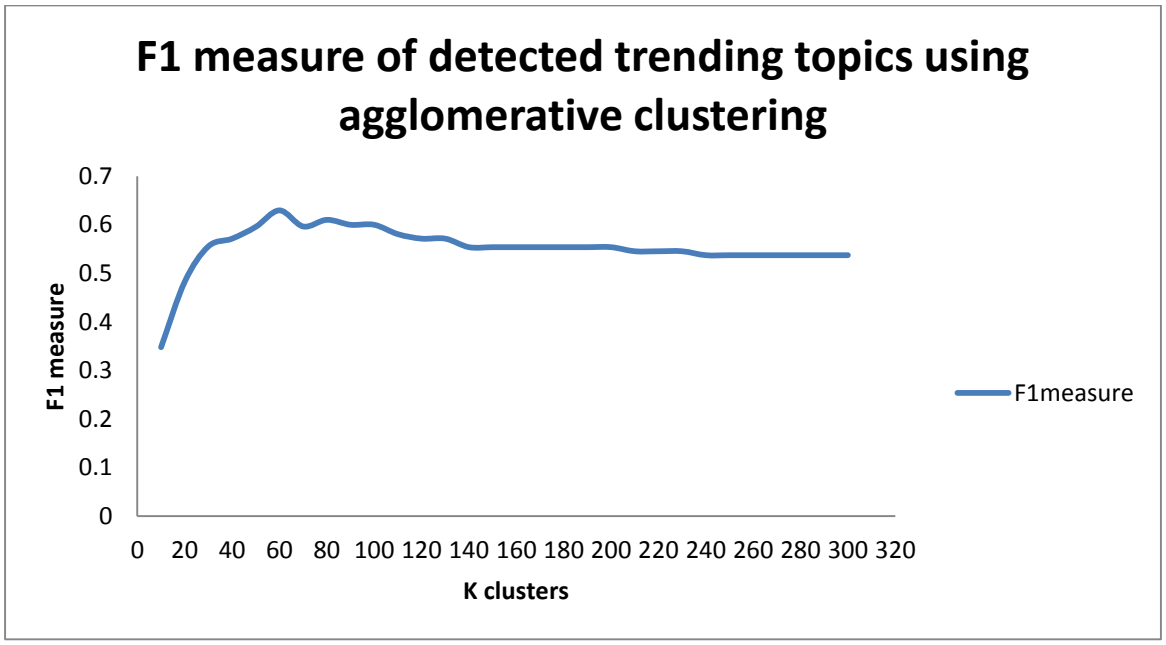

Figure 4-1 F1 measure of detected trending topics using agglomerative clustering

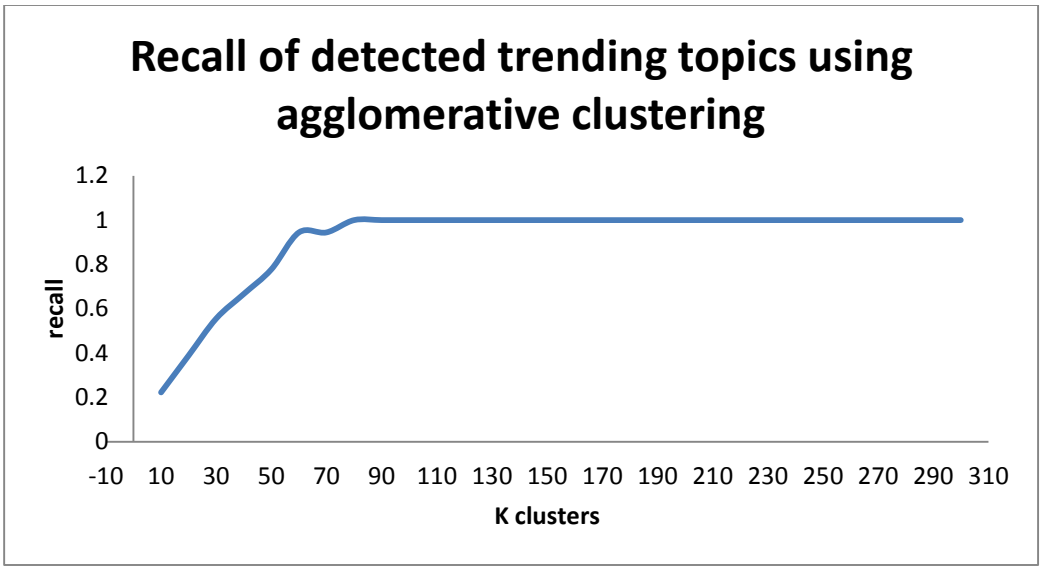

Figure 4-2 Recall of detected trending topics using agglomerative clustering

The highest F1 measure is recorded at k=60, and the recall reached 100% at k=80.

4.1.3.2. Topic extraction results

For k=60 and k=80, topic extraction method is applied. For every cluster the trending hash-tags are extracted to represent the topics. The results are evaluated against the annotated trending topics.

Fig (4-3) shows F1 measure values for extracted hash-tags using different frequencies in a cluster.

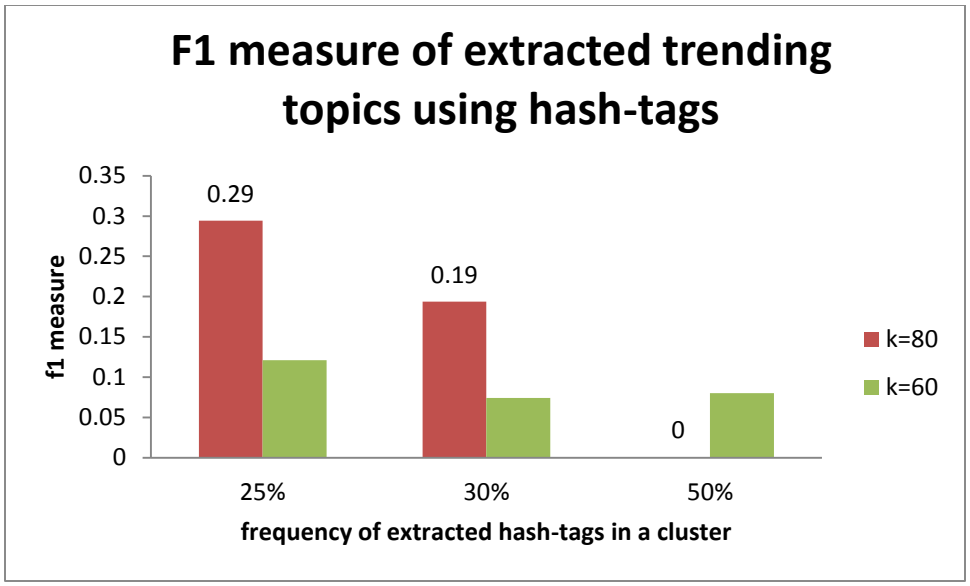

Figure 4-3 F1 measure of extracted trending topics using hash-tags

Fig (4-4) shows the recall values for trending topics using hash-tags of different frequencies in a cluster.

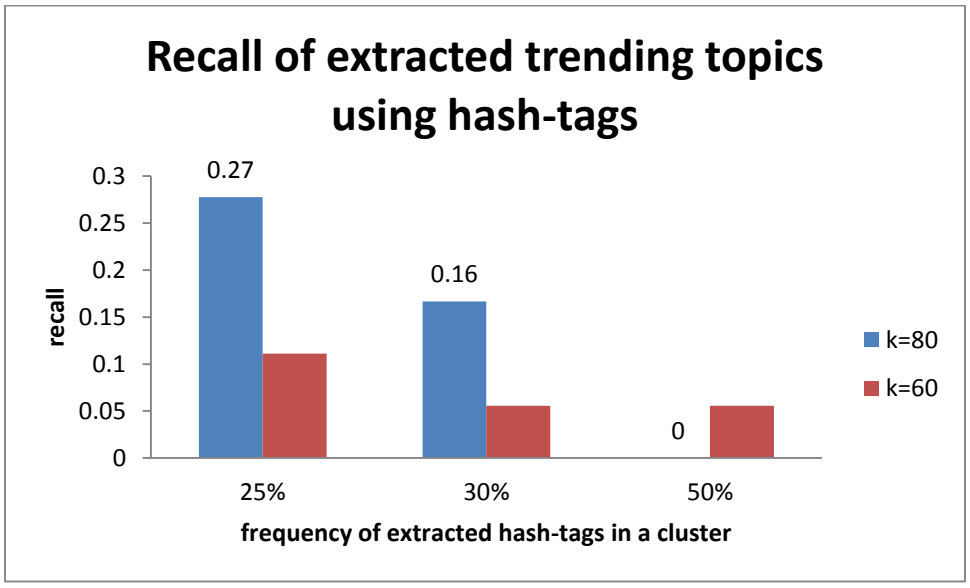

Figure 4-4 Recall of extracted trending topics using hash-tags

4.1.4. Discussion

From the above experiments we could find the highest F1 measure for clustering experiments is at k=60.

The recall reaches 100% at k=80, the 18 trending topics could be detected.

As it was expected, purity increases as k increases, because when the number of clusters increases the sizes of clusters decreases as well, so the percentage of tweets of belonging to the same topic in a cluster increases. The average intra-similarity of clusters increases as k increases as well.

Entropy decreases as k increases, as the more the close the tweets to each other in a cluster the more they are distant from other clusters.

We extracted hash-tags from each cluster to represent the topic of the cluster. We used the clustering solution at k=60 where the highest F1 measure value was recorded, and at k=80 where the 18 trending topics could be detected giving a recall of 100%. From each cluster the hash-tags occur more than or equal to 50%, 30% and 25% were extracted, each frequency in a separate experiment. The results showed that extracting hash-tags occur more than or equal to 25% of the cluster size at k=80 could achieve a recall of 0.27778.

In the following experiments we are going to investigate the effect of different factors on the extraction of trending topics.

4.2. Investigating different clustering techniques

4.2.1. Objective

In this experiment we are investigating the impact of different clustering techniques and how this affects the extraction of trending topics using hash-tags.

4.2.2. Method

To achieve our objective the following is performed:

- Tweets are represented using tf-itf
- Tweets are clustered using three different clustering techniques; k-means, repeated bisecting k-means (rb), and biased agglomerative clustering (bagglo).
- The results are evaluated against the baseline and the annotated topics in the same manner we used in the baseline.
- The best technique is then used, and topic extraction using hash-tags is applied, then the results are evaluated against the annotated topics, and the baseline.

4.2.3. Results

4.2.3.1. Clustering results

Fig (4-5) shows the F1 measure of the clustering techniques against the baseline.

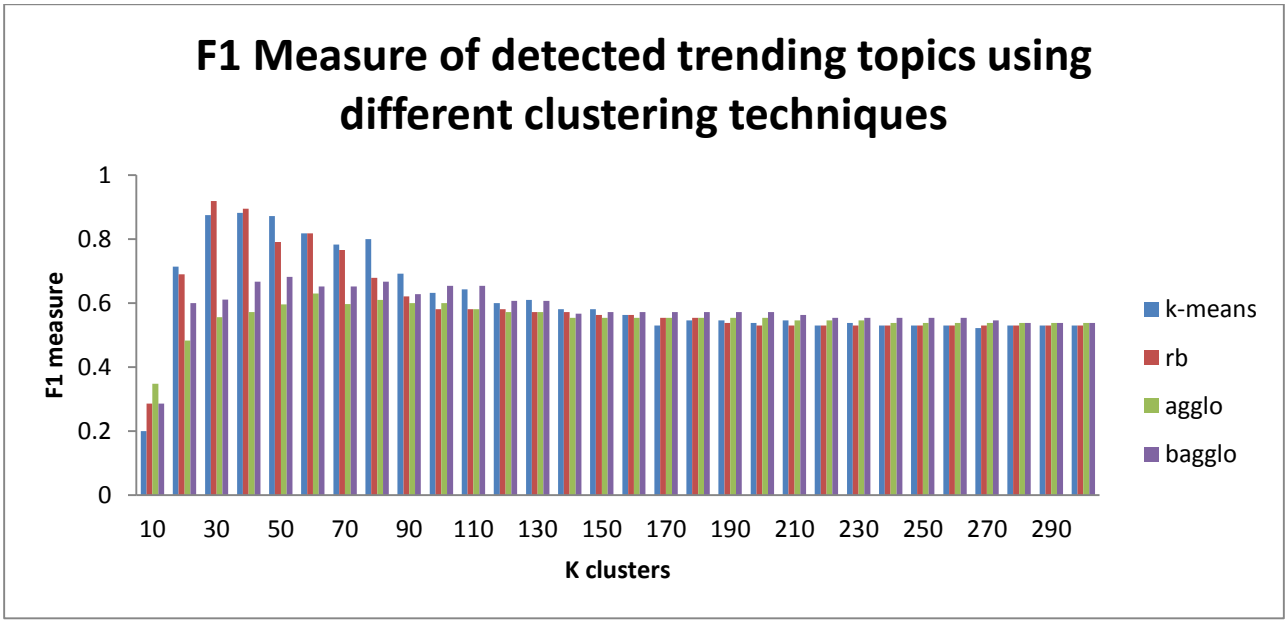

Figure 4-5 F1 measure of detected trending topics using different clustering techniques

Fig (4-6) shows the recall values of detected trending topics from each clustering technique against the baseline.

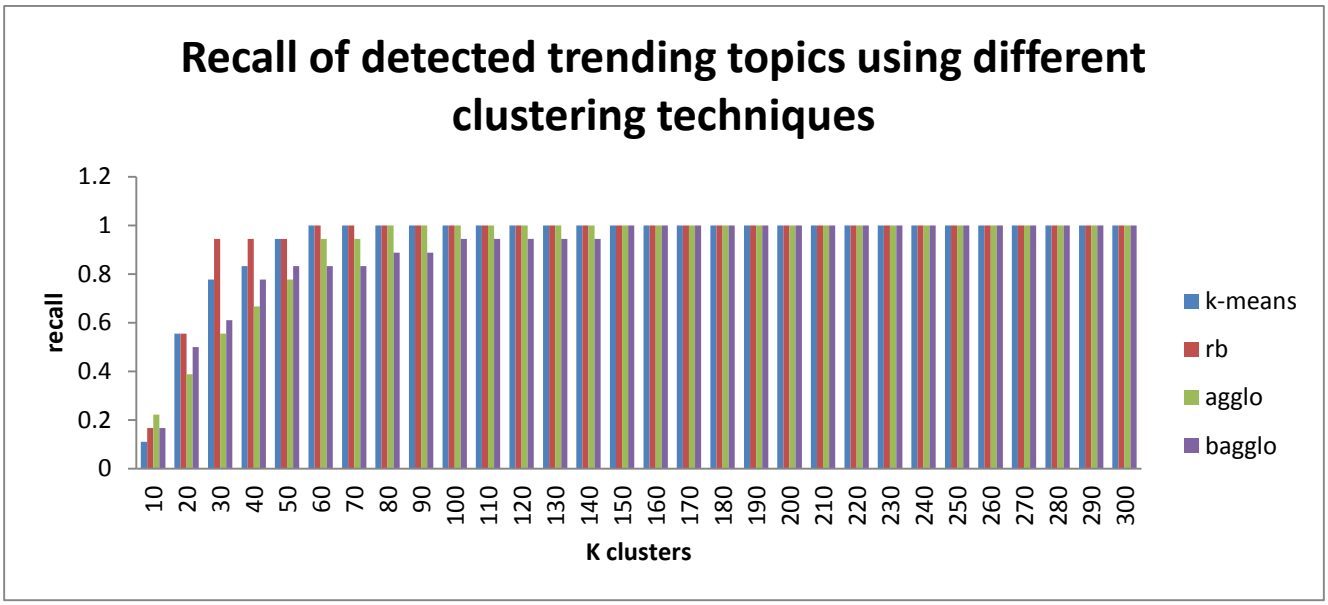

Figure 4-6 recall of detected trending topics using different clustering techniques

From the above results we could find that the recall reaches 100% at k=60 using k-means and repeated bisecting k-means, also the F1 measure values for both techniques are equal at the same k value.

Fig (4-7) shows the average F1 measure and recall values of detected trending topics using different clustering techniques.

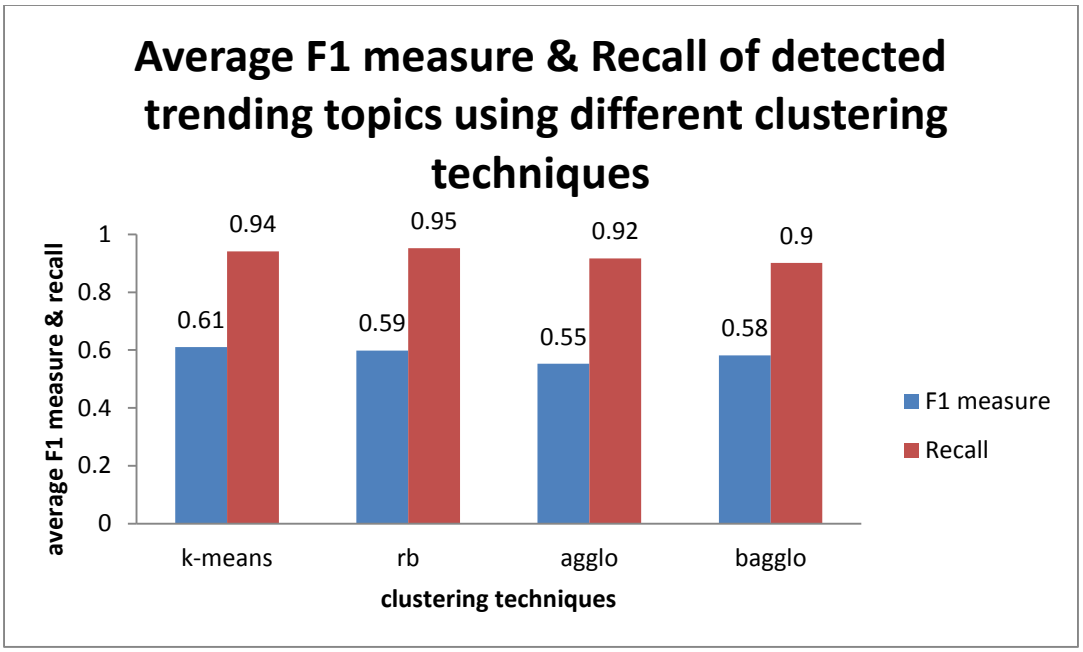

Figure 4-7 Average F1 measure and recall of detected trending topics using different clustering techniques

From the above graph we can deduce that the highest average F1 measure was recorded using k-means techniques, while the highest recall value was recorded using repeated bisecting k-means.

4.2.3.2. Topic extraction results

Topic extraction method using hash-tags are applied on both techniques at k=60, the F1 measure and recall values are shown in the figures (4-8) and (4-9)

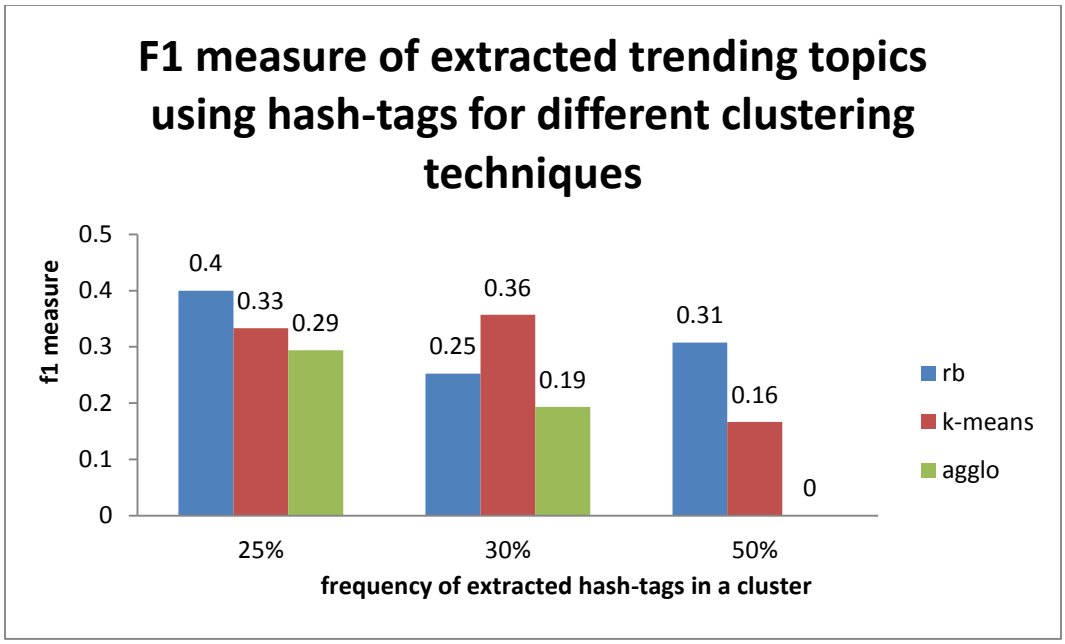

Figure 4-8 F1 measure of extracted trending topics using hash-tags for different clustering techniques

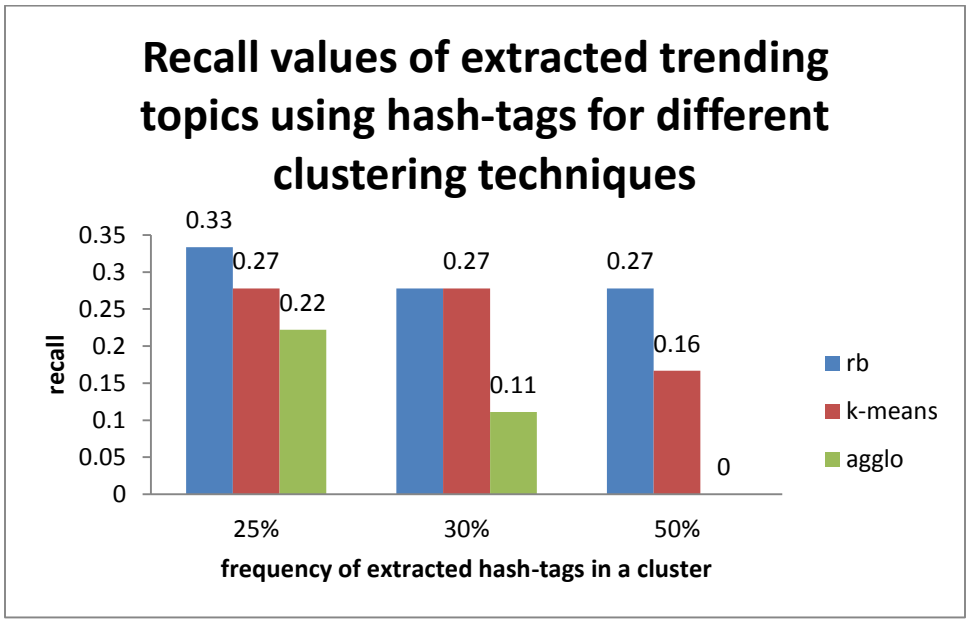

Figure 4-9 Recall of extracted trending topics using hash-tags for different clustering techniques

4.2.4. Discussion

From the above results we could find that the highest F1 measure was recorded at k= 30 using repeated bisecting k-means clustering technique. While the recall reached 100% at k=60 using k-means and bisecting k-means clustering techniques. Also the highest average F1 was recorded using k-means techniques while the highest average recall was recorded using repeated bisecting k-means technique.

From these observations we can deduce that k-means and repeated bisecting k-means performs better than the agglomerative techniques, as they result in higher F1 measures than both agglomerative and biased agglomerative clustering techniques. These results are consistent with what is known in the literature that hierarchical clustering lacks robustness and more sensitive to noise, as once an object is clustered it is not considered again which leaves no room for correcting errors that may occur in the beginning of the clustering by assigning an object to improper cluster. Also its computational complexity is at least $O(n^2)$ which is not suitable for dealing with very large data sets.

By comparing the average time, the average entropy and the entropy at k=60 for both the k-mean and the repeated bisecting clustering techniques we found the following in table (4-2)

Table (4-2) Average time, average entropy and entropy at k=60 for k-means and repeated bisecting k-means techniques

Clustering technique	Average time	Average entropy	Entropy at k=60
k-means	2.707033	0.1629	0.211
Repeated bisecting k-means	0.757233	0.13069	0.18

By applying topic extraction using hash-tags at k=60 using k-means, repeated bisecting k-means, and evaluate the results against the annotated topics and the baseline results, we found that using repeated bisecting k-means could achieve the highest recall when extracting hash-tags occur more than or equal to 25% of the cluster size.

From the previous observations we found that using repeated bisecting k-means at k=60 and extracting hash-tags occur more than or equal to 25% of the cluster size is the best combination so far to achieve our objective. This we will be calling baseline-1.

4.3. Investigating impact of feature representation

4.3.1. Objective

In this experiment we are investigating the impact of different representation of features and how it affects the extraction of trending topics.

4.3.2. Method

In order to achieve the objective the following is performed:

- N-grams; unigrams, bigrams and trigrams are extracted from the tweets. N-grams that occur more than 10 times in the tweets are included in the feature list. The vector representation for each tweet is composed of how frequent is each n-gram in the tweet.
- The tweets are clustered using repeated bisecting k-means technique.
- The tweets are again represented by using tf-itf of each n-gram.
- The tweets are then clustered using repeated bisecting k-means technique.
- The results of each representation are evaluated against the annotated topics and the results of repeated bisecting k-means using tf-itf representation.
- The topic extraction method is applied on the best clustering solution; the results are evaluated against the annotated topics and the results of repeated bisecting k-means using tf-itf.

4.3.3. Results

4.3.3.1. Clustering results

Fig (4-10) shows the F1 measures results from using N-grams features, N-grams-itf, and the baseline after changing clustering technique as described in the previous section..

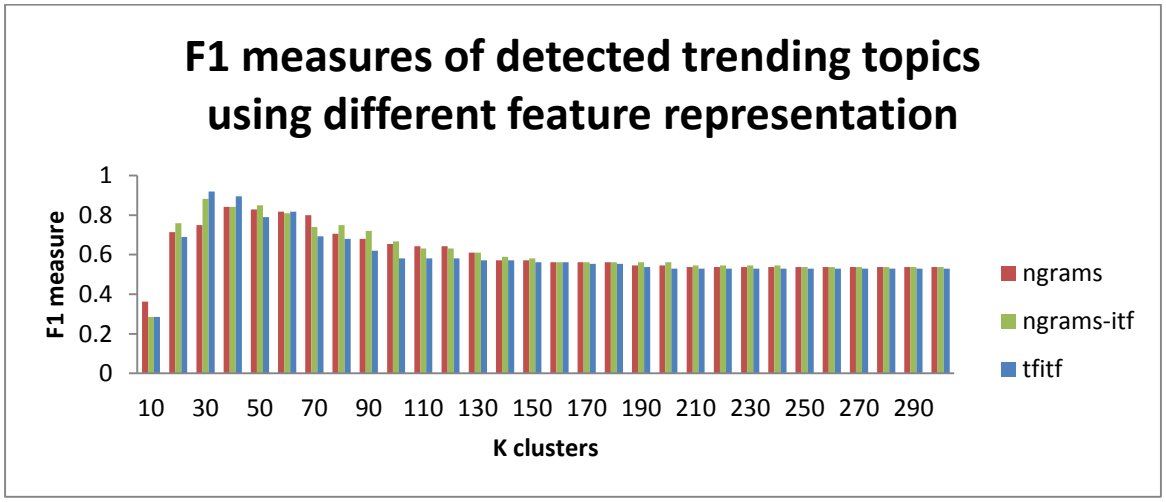

Figure 4-10 F1 measure of detected trending topics using different feature representation

Fig (4-11) shows the recall value of extracted trending topics using N-grams features, N-grams-itf and the baseline.

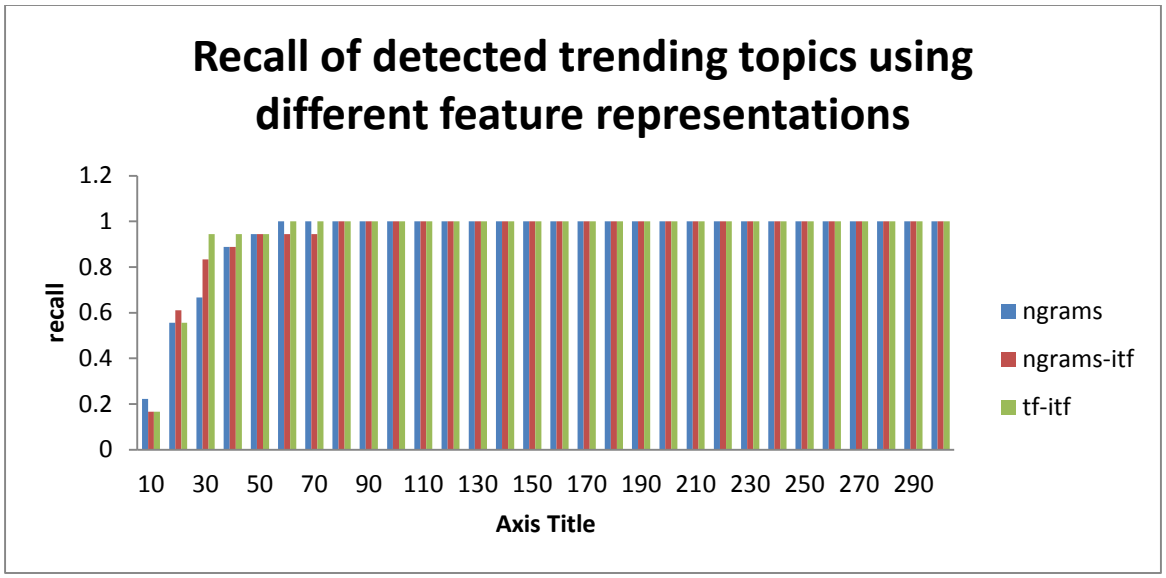

Figure 4-11 Recall of detected trending topics using different feature representations

From the above results we could find that using n-grams representation is equivalent to using tf-itf representation. They both reached a recall value of 100% at k=60.

4.3.3.2. Topic extraction results

Topic extraction method using hash-tags is applied on clustering solutions at k=60 using both representations. Fig (4-12) and Fig (4-13) show F1 measure and recall values of extracted trending topics.

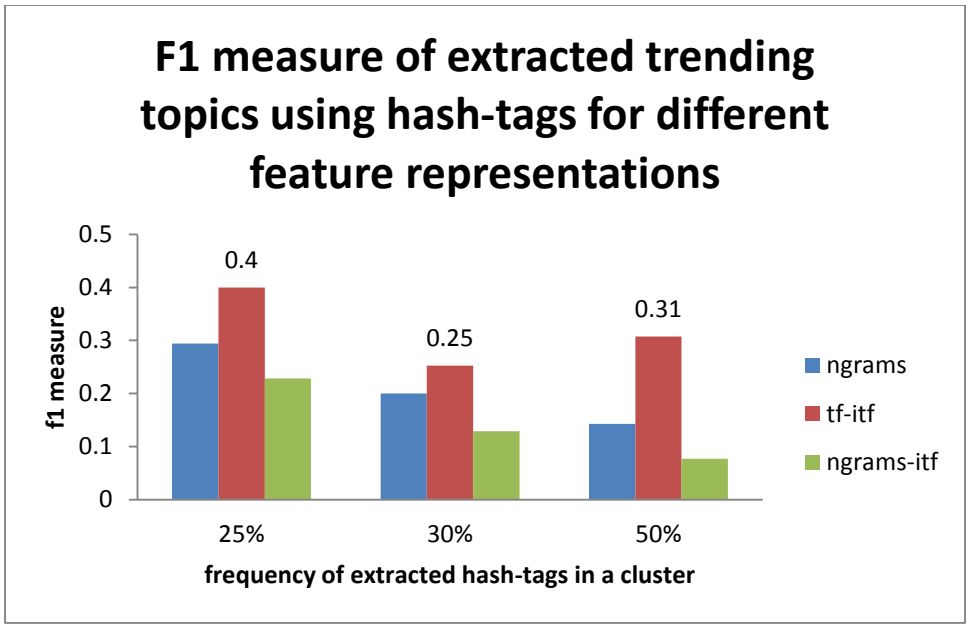

Figure 4-12 F1 measure of extracted trending topics using hash-tags for different feature representations

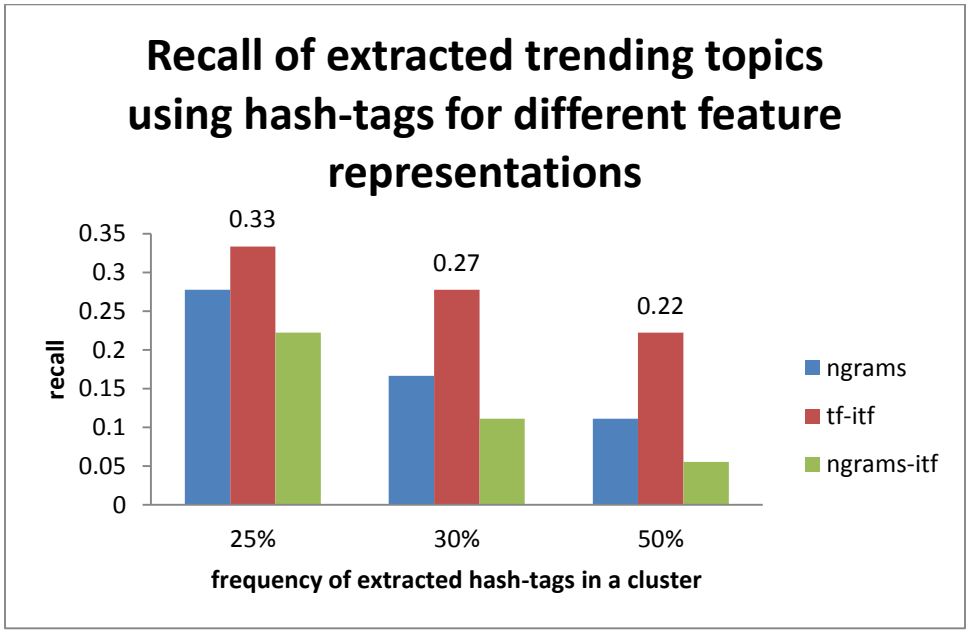

Figure 4-13 Recall of extracted trending topics using hash-tags for different feature representations

4.3.4. Discussion

From the above results of clustering we could observe that using tf-itf as features could record the highest F1 measure at k=30. Regarding the recall of trending topics, the results of using both tf-itf and the n-grams as features hit 100% at k=60.

Regarding the topic extraction results using hash-tags, we could find that using tf-itf representation achieved better results than using n-grams.

From the above observations we could find that using n-grams and n-grams-itf didn't improve the performance.

4.4.Investigating different topic extraction methods

4.4.1. Objective

In this experiment we are investigating applying different topic extraction methods to be able to extract the trending topics.

4.4.2. Method

In order to achieve our objective, the following is performed:

- Tweets are represented using tf-itf
- Clustered using repeated bisecting k-means technique at k=60.
- N-grams are extracted from each cluster to represent the trending topics.

  The following experiments are performed to determine the best combination of n-grams that is able to extract the topic.

  - Bigrams that occur more than or equal to 50% of the cluster size are extracted (bi50).
  - Bigrams that occur more than or equal to 30% of the cluster size are extracted (bi30).

- Unigrams that occur more than or equal to 50% of the cluster size and are not included in any bigram are extracted alongside with the best extracted bigrams (uni50).
- Unigrams that occur more than or equal to 30% of the cluster size and are not included in any bigram are extracted alongside with the best extracted bigrams (uni30).
- Unigrams that occur more than or equal to 25% of the cluster size and are not included in any bigram are extracted alongside with the best extracted bigrams (uni25).
- The best combination of unigrams and bigrams is determined.
- Trigrams that occur more than or equal to 50% are extracted alongside with the best combination of unigrams and bigrams not included in any trigram (tri50).
- Trigrams that occur more than or equal to 30% are extracted alongside with the best combination of unigrams and bigrams not included in any trigram (tri30).
- Trigrams that occur more than or equal to 25% are extracted alongside with the best combination of unigrams and bigrams not included in any trigram (tri25).

- The results are then evaluated against the annotated topics and against baseline1 where the extraction method is using hash-tags.

4.4.3. Results

We performed 10 experiments to choose the best extracted combination of unigrams and bigrams.

Fig (4-14) shows the F1 measures of extracted trending topics by extracting different unigrams and bigrams from a cluster.

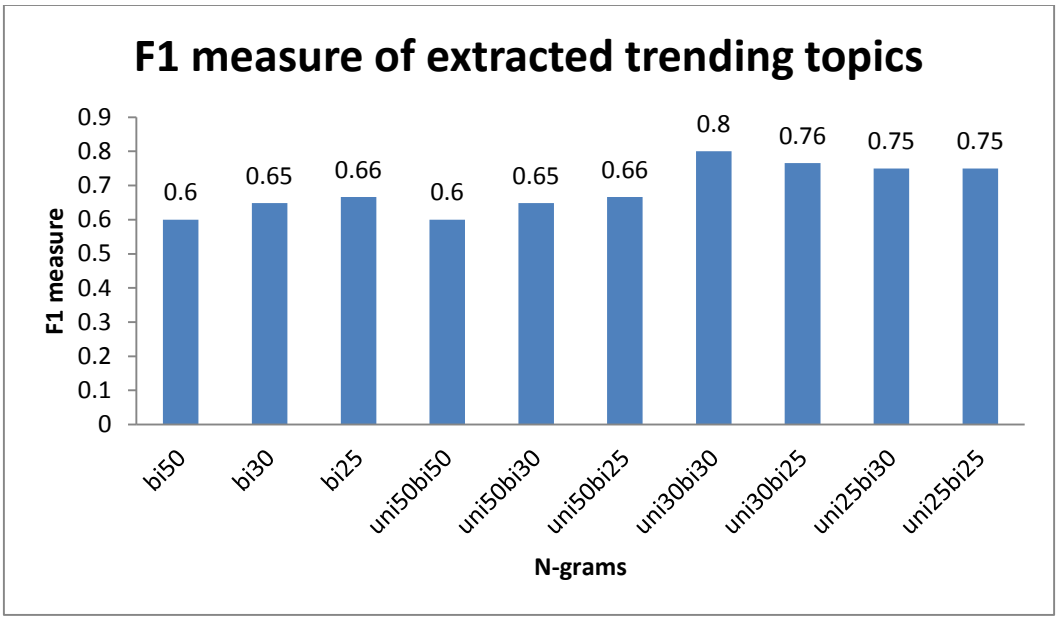

Figure 4-14 F1 measure of extracted trending topics using n-grams

Fig (4-15) shows the recall values of the extracted trending topics by extracting different unigrams and bigrams from a cluster.

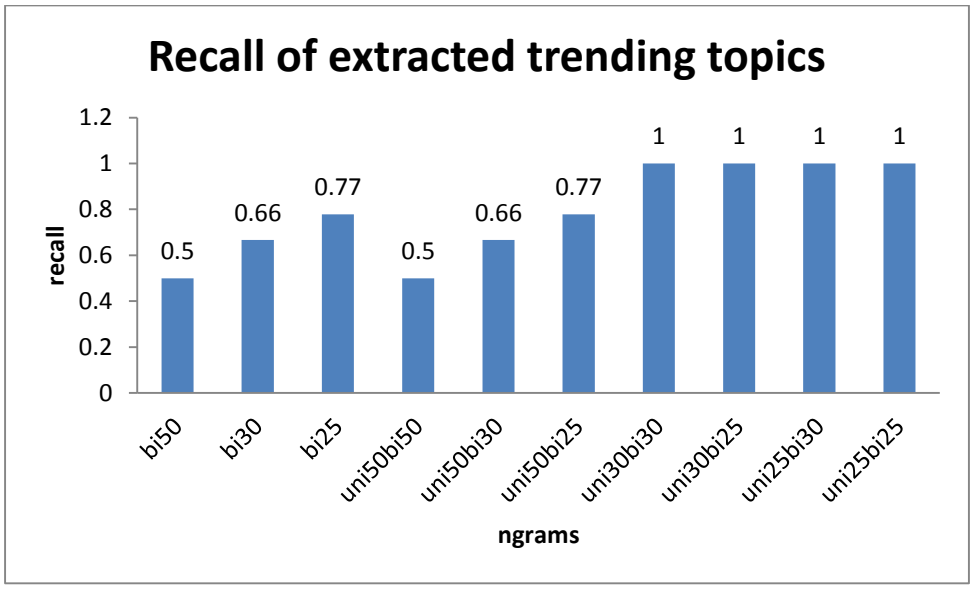

Figure 4-15 Recall of extracted trending topics using n-grams

We chose the combination of extracting unigrams and bigrams those occur more than or equal to 30% of the cluster size.

Trigrams of different frequencies are extracted alongside with the unigrams and bigrams combination. Three experiments were performed for trigrams occur more than or equal to 25%, 30%, and 50% of the cluster size. Trigrams are first extracted, then bigrams not included in the trigrams are also extracted, then unigrams not included in both bigrams and trigrams are extracted.

Fig (4-16) shows the F1 measures.

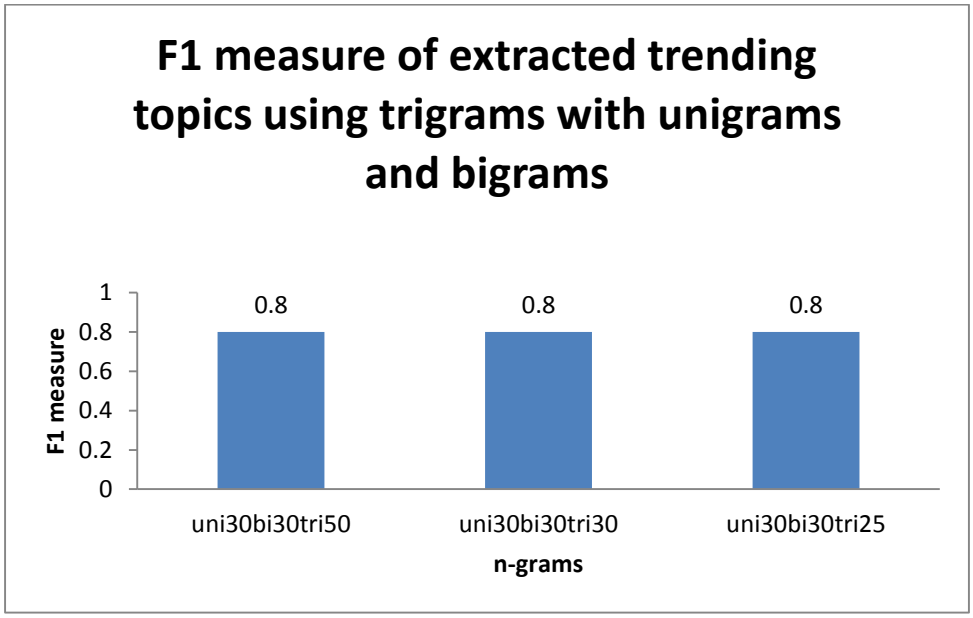

Figure 4-16 F1 measure of extracted trending topics using trigrams with unigrams and bigrams

Fig (4-17) shows the recall values of extracting different trigrams frequencies alongside with unigrams and bigrams.

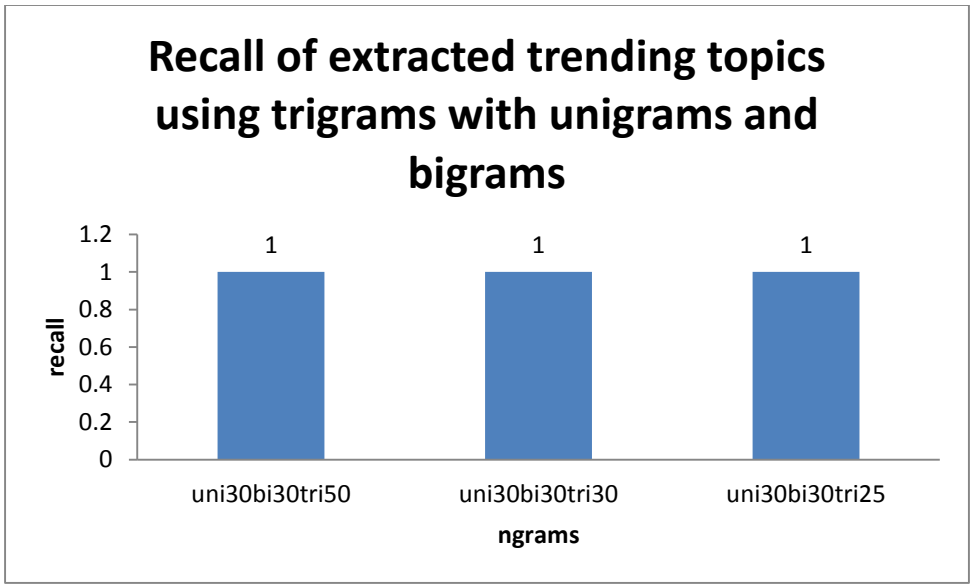

Figure 4-17 Recall of extracted trending topics using trigrams with unigrams and bigrams

Fig (4-18) shows the F1 measure, and recall values of topic extraction method using n-grams and using hash-tags.

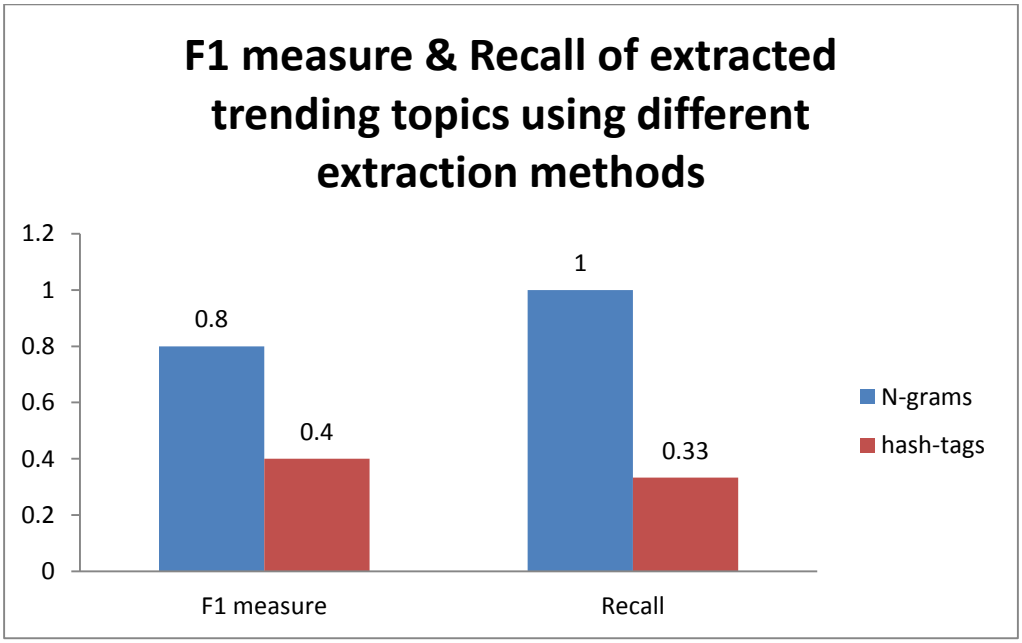

Figure 4-18 F1 measure and recall of extracted trending topics using different extraction methods

4.4.4. Discussion

From the above results we could determine the best combination of n-grams that extracts trending topics in a way satisfying our objective.

Extracting trigrams, bigrams and unigrams each occur more than or equal to 30% of the cluster size is found to be the best combination.

Extracting trigrams didn't enhance the F1 measure or recall values but it enhanced the quality of the results, as trigrams are more meaningful.

We could deduce that topic extraction method using N-grams is achieving better results than using hash-tags.

Finally we can deduce that using tf-itf feature representation, repeated bisecting k-means, and applying topic extraction method using extracted N-grams is the best combination achieving our objective. This we will be calling baseline-2 so we can compare further results to it.

Chapter 5. Trending Topic Extraction using Feature-Pivot Approach

In this chapter we are investigating how applying the feature-pivot approach on the baseline data will affect the extraction of trending topics for a twitter user.

The feature-pivot algorithm is based on extracting trending unigrams then grouping them together to represent a topic.

The algorithm we are using to group the trending unigrams is called content similarity. It checks if unigrams related to the same topic by checking the unigrams co-occurring with them. A pair of unigrams are said to be related if the number of unigrams co-occurring along with them exceeds a certain threshold.

The algorithm of content similarity goes over two levels. The first one checks if a pair of unigrams related to the same topic when the number of their common co-occurring unigrams exceeds a certain threshold. The second one checks if the second unigram in the pair and other unigrams related to the topic when the number of their common co-occurring unigrams exceeds a certain threshold.

First we are investigating the effect of different values of the threshold of the first level of content similarity and how it affects the results.

Then we are investigating the effect of different values of the threshold of the second level of content similarity and how it affects the results.

Finally we apply the document pivot and the feature pivot approaches to different data sets, of different sizes and from different domains to validate our results using the two-sample paired significance t-test.

5.1. Investigating different values of the threshold of the first level of content similarity

5.1.1. Objective

In this experiment we are investigating how different values of the threshold used to determine if a pair of unigrams related to the same topic (the first level of content similarity (θ_3)) affect the results of extracting trending topics.

5.1.2. Method

In order to achieve the objective the following is performed:

1. Apply feature-pivot approach on the preprocessed tweets of the baseline data by doing the following:
    a. Extract the set of unigrams occur more than 10 times in the data set.
    b. Extract the significant unigrams occurring with a frequency exceeds the average frequency of the set of unigrams.
    c. Then extract the associated set of tweets for each significant unigrams where they occur.
    d. From each set of tweets the set of frequent common unigrams is extracted where their frequency exceeds the proportional frequency of the unigrams in the set of tweets.
    e. If the number of tweets in a topic is 20 so this topic is considered trending.
    f. Set the value of the threshold of the first level of content similarity (θ_3) to different values: 0.1, 0.2, 0.3, 0.4, and 0.5, while setting the value of the second level of content similarity (θ_4) to an arbitrary value which is 0.45.
2. Evaluate the results against the annotated data to get the recall and F1 measure.
3. Determine the value of the threshold that achieved the highest recall and F1 measure.

5.1.3. Results

We performed 5 experiments to determine the best value of the threshold of the first level of content similarity.

Figure (5-1) shows the recall, and F1 measure values of different values for the threshold.

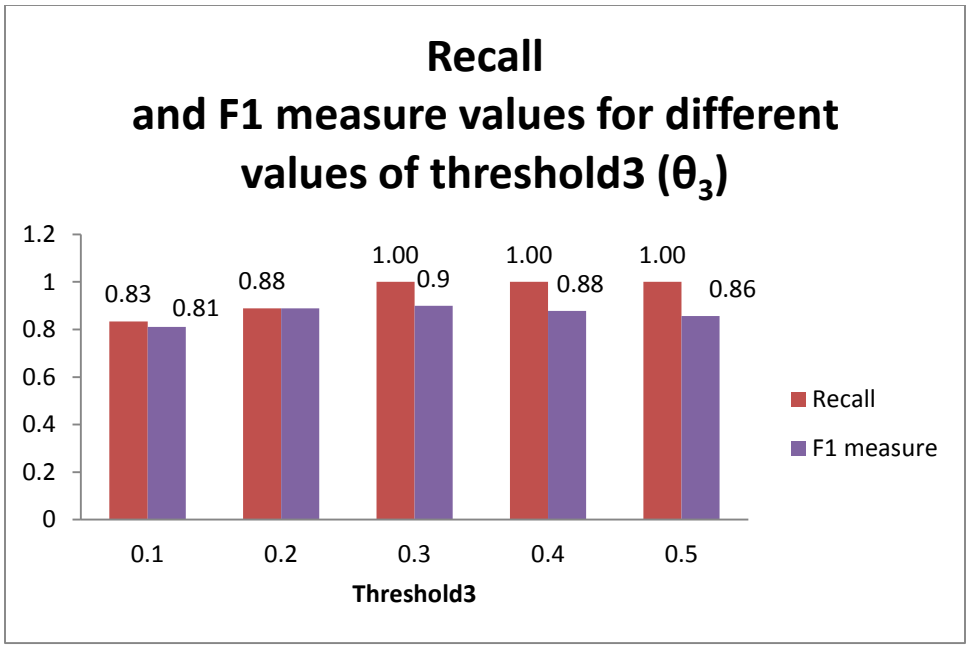

Figure 5-1 Recall and F1 measure values for different values of θ3

5.1.4. Discussion

From the previous experiments we could find that the recall reached 100% at values of θ_3 at 0.3, 0.4 and 0.5, while the F1 measure reached its highest value of 0.9 at the value of 0.3

From this we choose the value of θ_3 to be 0.3 where the highest recall and F1 measure values were recorded.

5.2. Investigating different values of the threshold of the second level of content similarity

5.2.1. Objective

In this experiment we are investigating how different values of the threshold used to determine if further unigrams are related to the topic (the second level of content similarity (θ_4)) affect the results of extracting trending topics.

5.2.2. Method

In order to achieve the objective the following is performed:

1. Apply feature-pivot approach on the preprocessed tweets of the baseline data by doing the following:
    a. Extract the set of unigrams occur more than 10 times in the data set.
    b. Extract the significant unigrams occurring with a frequency exceeds the average frequency of the set of unigrams.
    c. Then extract the associated set of tweets for each significant unigrams where they occur.
    d. From each set of tweets the set of frequent common unigrams is extracted where their frequency exceeds the proportional frequency of the unigrams in the set of tweets.
    e. The number of tweets in a topic is set to 20 so this topic is considered trending.
    f. Setting value of the threshold of the second level of content similarity (θ_4) to different values: 0.1, 0.2, 0.3, 0.4, and 0.5, while setting the value of the first level of content similarity (θ_3) to 0.3 as determined from the previous experiment.
2. Evaluate the results against the annotated data to get the recall and F1 measure.
3. Determine the value of the threshold that achieved the highest recall and F1 measure.

5.2.3. Results

We performed 5 experiments to determine the best value of the threshold of the second level of content similarity.

Figure (5-2) shows the recall and F1 measure values of different values for the threshold.

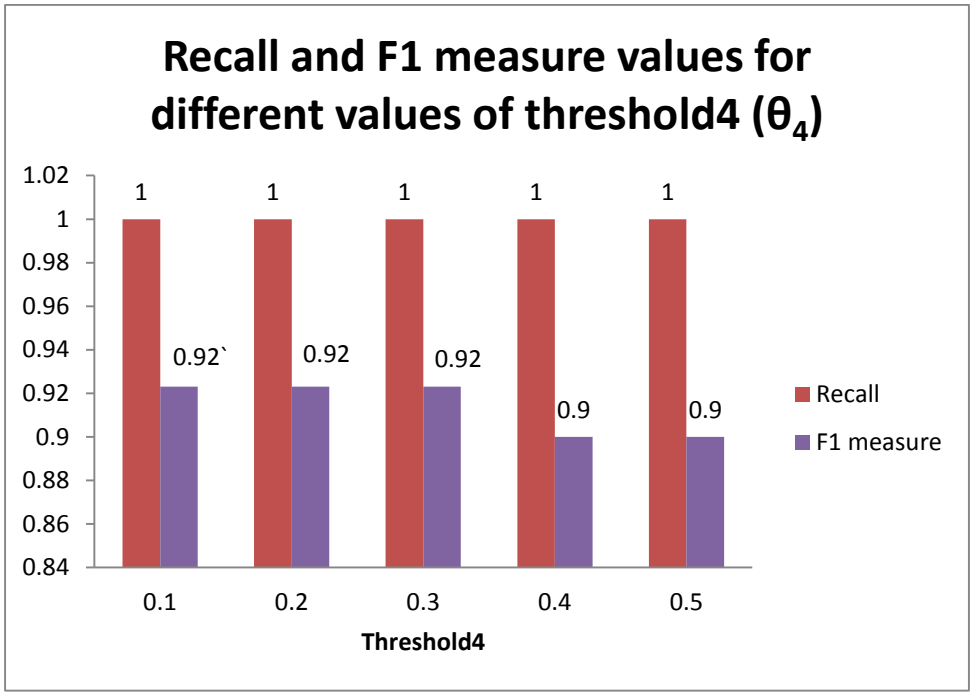

Figure 5-2 Recall and F1 measure values for different values of θ4

5.2.4. Discussion

From the above results we could observe that the recall reached 100% for all values of the threshold (θ_4). The F1 measure gave the highest value of 0.92307 at threshold values of 0.1, 0.2, and 0.3.

We will pick the value of 0.2 as an average value of the three values 0.1, 0.2, and 0.3.

From the above two experiments we can deduce that the value of threshold of the first level of content similarity (θ_3) is 0.3 and the value of the threshold of the second level of content similarity (θ_4) is 0.2.

Figure (5-3) shows the recall and F1 measure values resulted from applying the document pivot approach and the feature pivot approach on the same data set.

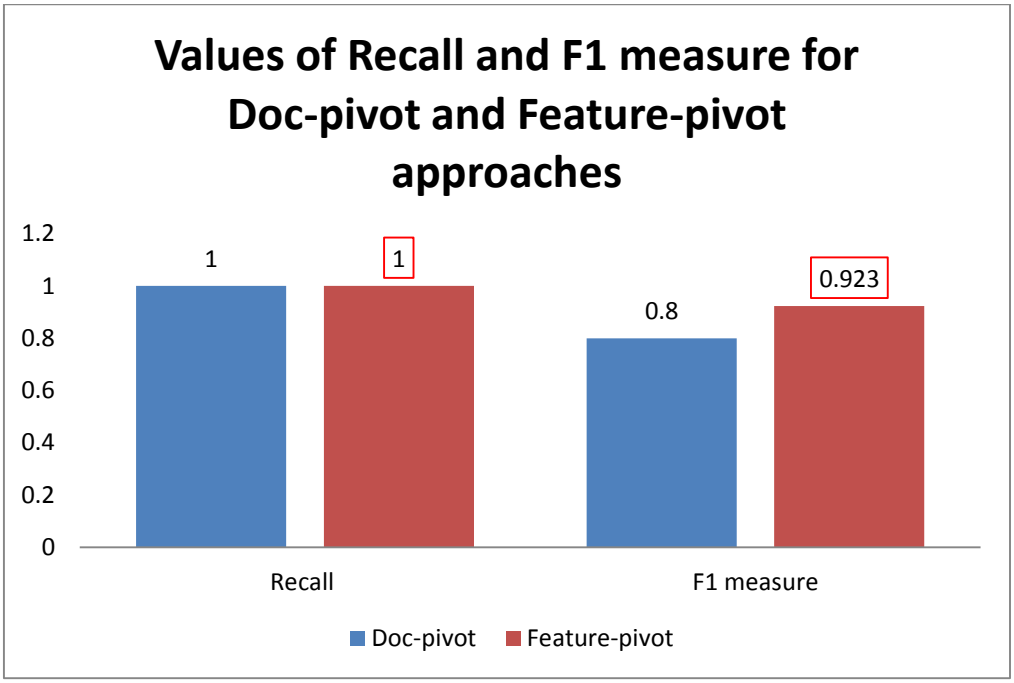

Figure 5-3 Values of Recall and F1 measure for Doc-pivot and Feat-pivot approaches

The figure shows by applying the feature pivot approach we could achieve a F1 measure of 0.923 in contrast with a value of 0.8 resulted from applying the document pivot approach.

To validate that the feature pivot approach is performing better we are performing the experiments in the following section.

5.3. Applying both Doc-pivot and Feature-pivot approaches on different data sets

In this experiment we are applying both approaches on different data sets of different sizes and from different domain to find how significant the difference between applying both approaches is.

5.3.1. Objective

The objective of this experiment to examine whether there is statistical significance between results achieved from applying both approaches on different data sets.

5.3.2. Method

In order to achieve the objective of this experiment we are performing the following:

1. Collect data of sizes 200,400,600, and 1200 tweets from three different domains; sports, entertainment, and news.
2. Annotate all data sets to determine trending topics in each set.
3. Preprocess all the data sets by removing stop words, punctuation marks, and account names.
4. Apply document pivot approach using repeated bisecting k-means at k=60 and topic extraction method using unigrams, bigrams and trigrams occurring more than or equal to 30% of the cluster size.
5. Validate the results against the annotated data and record the recall, precision and F1 measure values.
6. Apply feature pivot approach using α at value of 20, θ_3 at value of 0.3 and θ_4 at value of 0.2
7. Validate the results against the annotated data and record the recall, precision and F1 measure values.
8. Apply Two-sample paired significance t-test on the recall, precision and F1 measure values recorded by each approach and record its significance.

5.3.3. Results

We performed 12 experiments; 4 different sizes 200,400,600, and 1200 tweets from 3 domains; sports, entertainments, and news.

Figure (5-4) shows the recall values of each experiment, and figure (5-5) shows the mean of the recall values result from applying both approaches.

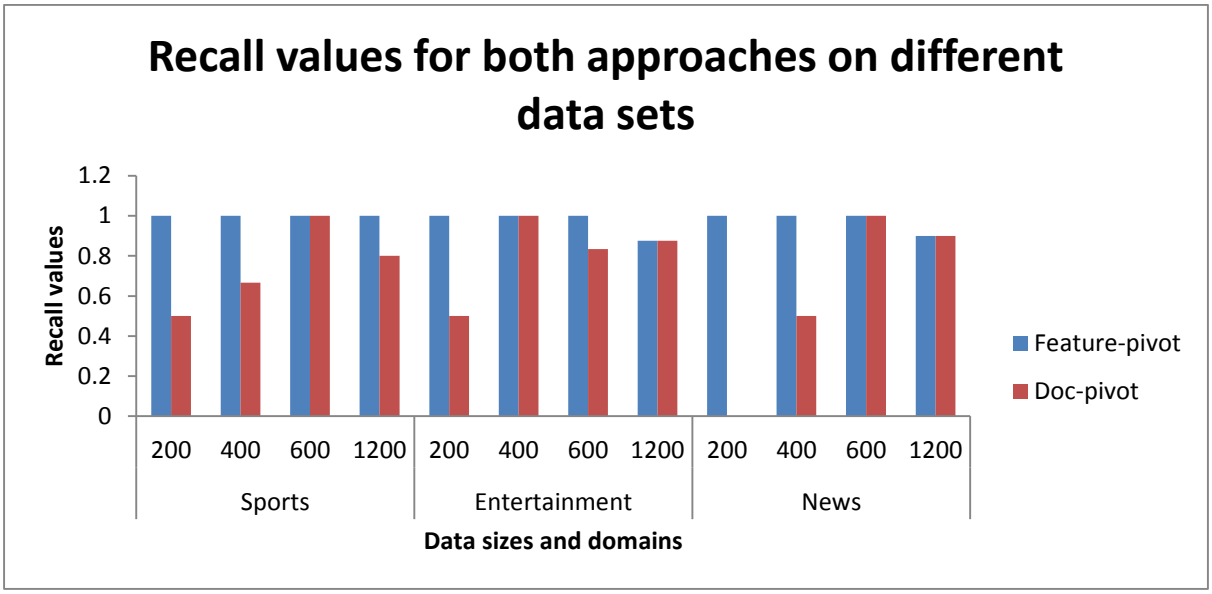

Figure 5-4 Recall values for both approaches on different data sets

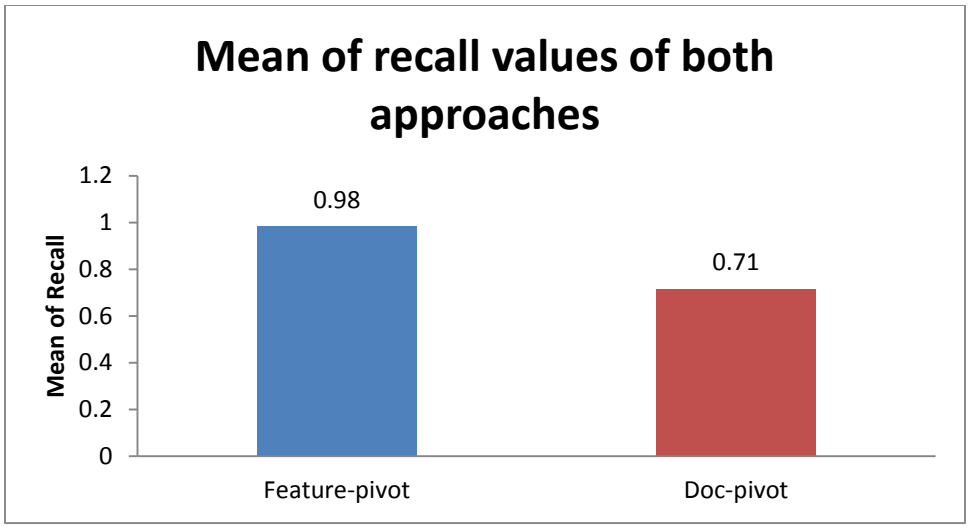

Figure 5-5 Mean of recall values of both approaches

Figure (5-6) shows the precision values of each experiment, and figure (5-7) shows the mean of the precision values result from applying both approaches.

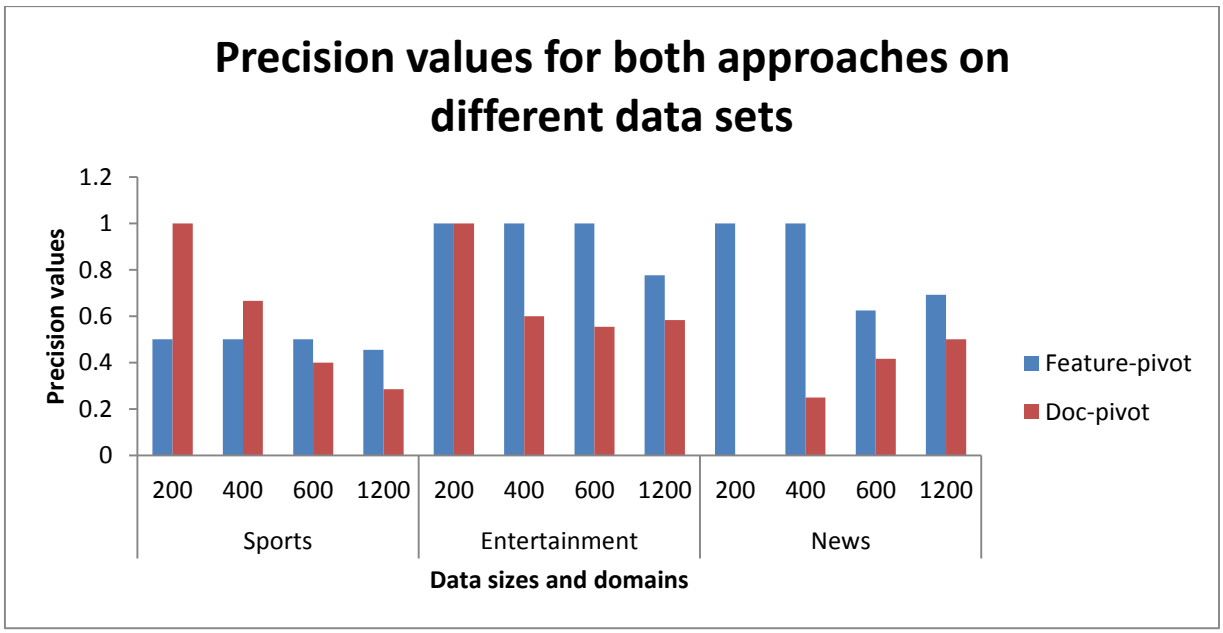

Figure 5-6 Precision values for both approaches on different data sets

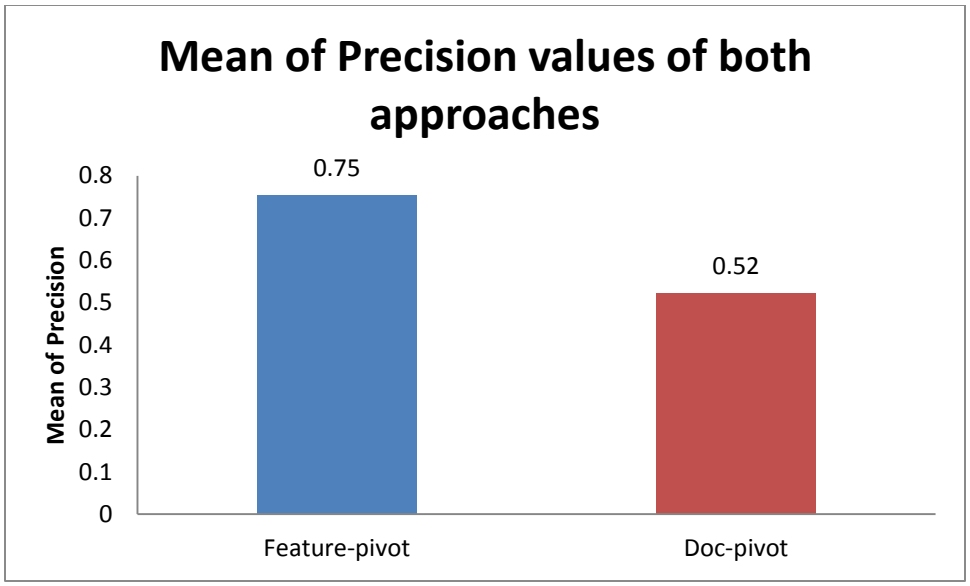

Figure 5-7 Mean of precision values of both approaches

Figure (5-8) shows the F1 measure values of each experiment, and figure (5-9) shows the mean of the F1 measure values result from applying both approaches.

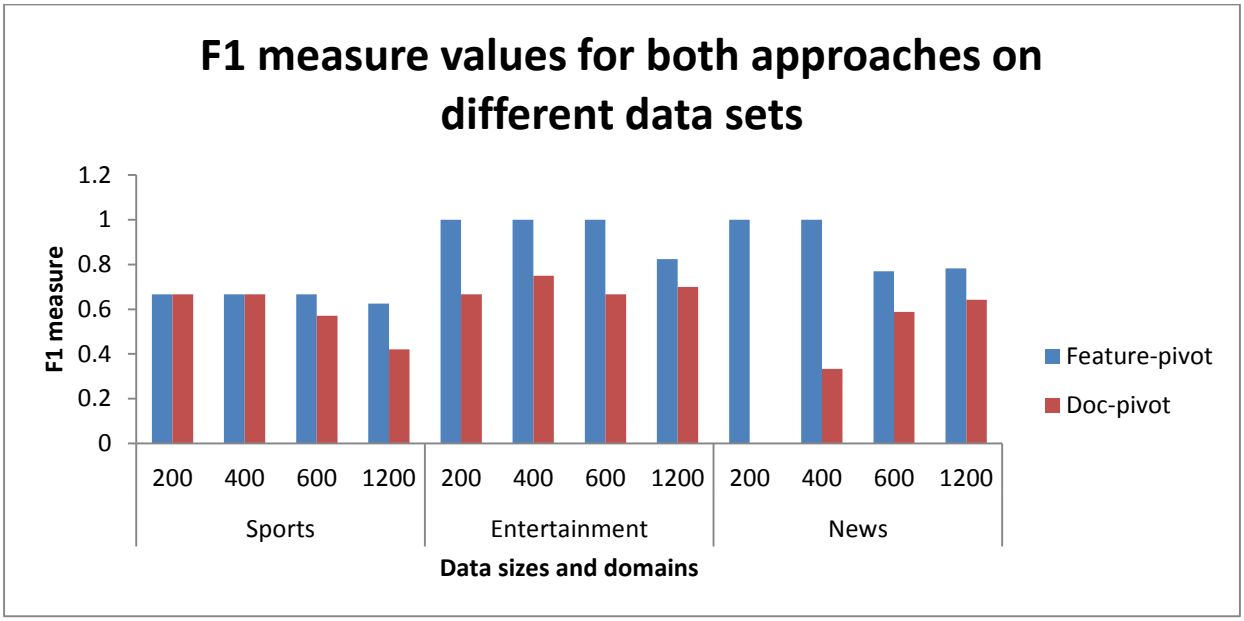

Figure 5-8 F1 measure values for both approached on different data sets

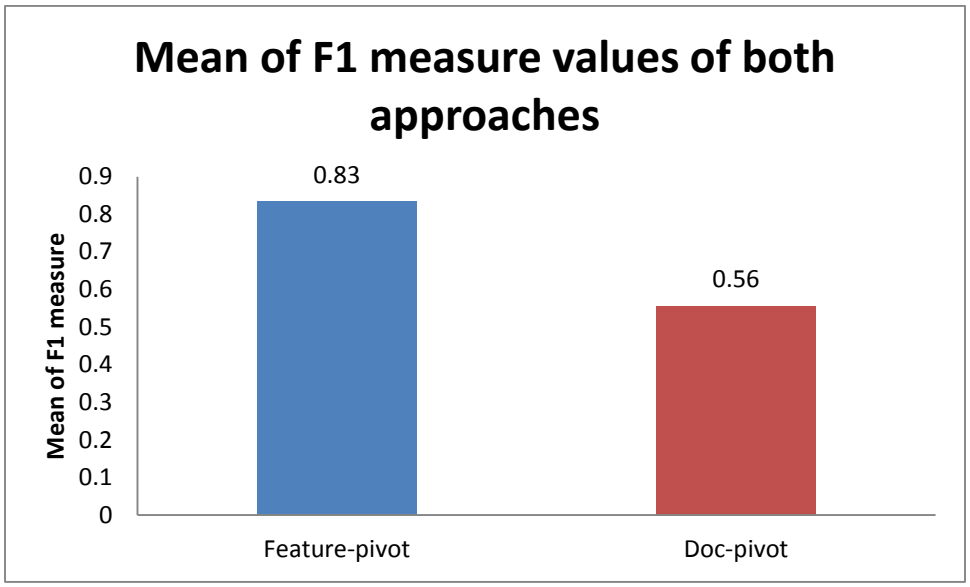

Figure 5-9 Mean of F1 measure values of both approaches

Since the mean of the values resulted from applying the feature pivot approach is greater than those resulted from applying the document pivot approach so we need to apply a One-tailed paired t-test.

Our hypothesis would be that there is an increase in performance yields from applying the feature pivot approach.

To get the value of $t_{obtained}$ (test statistic) we use the following formula:

$$t_{obtained} = \frac{\overline{D}}{\frac{S_D}{\sqrt{n}}}$$

Where *n* is the number of samples which is 12, $\overline{D}$ is the mean of difference between pairs, and S_D is the standard deviation of the difference between pairs.

By applying Two-sample one-tailed paired significance t-test at α =0.05 and a confidence level of 90% on the recall, precision, and F1 measure resulted from the above experiments we got the following results:

1. For Recall values:

Table 5-1 Summary of the Recall Results

Experiment Number	Recall using Feature-pivot	Recall using Document-Pivot	Difference D	Square difference D^2
1	1	0.5	0.5	0.25
2	1	0.666	0.334	0.111556
3	1	1	0	0
4	1	0.8	0.2	0.04
5	1	0.5	0.5	0.25
6	1	1	0	0
7	1	0.833	0.167	0.027889
8	0.875	0.875	0	0
9	1	0	1	1
10	1	0.5	0.5	0.25
11	1	1	0	0
12	0.9	0.9	0	0
Sum			3.201	1.929445
Mean	0.9813	0.7145	0.266725	
Standard Deviation	0.0436	0.298	0.312706	

We got a value of $t_{obtained}$ =2.954729, using a degree of freedom (*n-1*) which is 11, from the one-tailed t-test table at α=0.05 and a confidence interval of 90% we get $t_{critical}$ = 1.796, thus we got $t_{obtained} > t_{critical}$

Thus there was a significant difference in the recall values between applying the feature pivot approach and applying the document pivot approach at 90% confidence interval which proves our hypothesis.

2. For Precision values:

Table 5-2 Summary of Precision values

Experiment Number	Precision using Feature-pivot	Precision using Document-Pivot	Difference D	Square difference D^2
1	0.5	1	-0.5	0.25
2	0.5	0.666	-0.166	0.027556
3	0.5	0.4	0.1	0.01
4	0.4545	0.2857	0.1688	0.02849344
5	1	1	0	0
6	1	0.6	0.4	0.16
7	1	0.555	0.445	0.198025
8	0.777	0.5833	0.1937	0.03751969
9	1	0	1	1
10	1	0.25	0.75	0.5625
11	0.625	0.4166	0.2084	0.04343056
12	0.6923	0.5	0.1923	0.03697929
Sum			2.7922	2.3545
Mean	0.7541	0.5214	0.23268	
Standard Deviation	0.2349	0.2888	0.393678	

We got a value of $t_{obtained}$ =2.047457, using a degree of freedom (*n-1*) which is 11, from the one-tailed t-test table at α=0.05 and a confidence interval of 90% we get $t_{critical}$ = 1.796, thus we got $t_{obtained} > t_{critical}$

Thus, there was a significant difference in the precision values between applying the feature pivot approach and applying the document pivot approach at 90% confidence interval which proves our hypothesis.

For F1 measure values:

Table 5-3 Summary of F1 measure values

Experiment Number	F1 measure values using Feature-pivot	F1 measure values using Document-Pivot	Difference D	Square difference D^2
1	0.666	0.666	0	0
2	0.666	0.667	-0.001	0.000001
3	0.666	0.5714	0.0946	0.00894916
4	0.625	0.421	0.204	0.041616
5	1	0.666	0.334	0.111556
6	1	0.75	0.25	0.0625
7	1	0.6667	0.3333	0.11108889
8	0.8235	0.7	0.1235	0.01525225
9	1	0	1	1
10	1	0.333	0.667	0.444889
11	0.7692	0.5882	0.181	0.032761
12	0.7826	0.6426	0.14	0.0196
Sum			3.3264	1.8482133
Mean	0.8332	0.556	0.2772	
Standard Deviation	0.1575	0.2117	0.290162	

We got a value of $t_{obtained}$ =3.30935, using a degree of freedom (*n-1*) which is 11, from the one-tailed t-test table at α=0.05 and a confidence interval of 90% we get $t_{critical}$ = 1.796, thus we got $t_{obtained} > t_{critical}$

There was a significant difference in the F1 measure values between applying the feature pivot approach and applying the document pivot approach at 90% confidence interval which proves our hypothesis.

5.3.4. Discussion

From the above experiments we could deduce that applying the feature pivot approach achieved significantly better results than applying the document pivot approach. That was proved by applying both approaches on different data set sizes (200, 400, 600, and 1200) from different domains (sports, entertainment, and news). The Two-sample paired one-tailed significance test was applied to the values of the recall, precision and F1 measure resulted from applying both approaches on the data sets. The test showed that we could prove our hypothesis that applying the feature pivot approach achieves significantly better results.

This can lead us to the conclusion that applying the feature pivot approach achieves our objective of extracting trending topics for Egyptian Twitter user.

Chapter 6. Conclusion and future work

Twitter has become a very important source of information about the current events all around the world. The users of Twitter are increasing every day and the usage of Twitter in different domains is increasing as well. It has become part of the news media, advertising campaigns, business plans, social events, etc.

A Twitter user follows lots of accounts among them, public figure, news accounts, companies' accounts, and friends. In order to know what the people he/she follows discuss at any time, he/she has to go through all the posted tweets.

In our research, we are presenting an easier way for the user to know the trending discussed topics by account he follows without having to go through all the posted tweets.

To achieve our objective we applied the document pivot approach to cluster tweets belonging to the same topic together. Different clustering techniques were applied, from where we found that using repeated bisecting k-means could achieve the best results. Different feature representations were applied and we found that representing tweets using tf-itf could achieve the best results. To extract trending topics we applied two methods. The first one is by extracting the frequent hashtags that exceed a certain threshold from each cluster. And the second one is by extracting the frequent n-grams that exceeds a certain threshold from each cluster. We found that extracting trigrams, bigrams, and unigrams each occur more than or equal to 30% of the cluster size could achieve better results than using hash-tags. It could extract trending topic with a recall value of 100% and F1 measure of 0.8. On contrary using hash-tags achieved a recall value of 33% and F1 measure of 0.4.

By applying the feature pivot approach using content similarity algorithm we developed which is based on extracting significant unigrams occurring with a frequency more than or equal to the average frequency of all unigrams in the data set as features. Then group features related to the same topic by applying content similarity between tweets in which those features appear. The content similarity algorithm goes over two levels; the first one checks if a pair of two features related to the same topic that is if the number of common unigrams appear along with them both exceeds a certain threshold. The second level checks if further features related to the same topic

that is if the number of common unigrams appears with the second feature in the pair and other features exceed a certain threshold. By setting the threshold of the first level of content similarity to 0.3 and the second level to 0.2 we could achieve a recall value of 100% and F1 measure of 0.923 which is higher than that achieved by applying the document pivot approach.

To validate our results we applied both approaches on 12 different data sets. The data sets are of different sizes (200,400,600, and 1200) tweets and from three different domains; sports, entertainment and news. Then we applied the Two-sample paired one-tailed t-test to measure how significant are the results achieved by applying the feature pivot approach. The test showed that the feature pivot approach achieves better results at a confidence interval of 90% in extracting trending topics from twitter than applying the document pivot approach.

Our results look promising as for our knowledge extracting trending topics for a Twitter user was not tackled specially for an Egyptian user.

In our future work we will work on enhancing the results to get better precision values, and to implement a working web-based tool that can work near real time. We are considering different techniques like machine learning, deep neural network, fuzzy logic, and words embedding for extracting semantically related.

Chapter 7. References

1. Aiello, L.M.; Petkos, G.; Martin, C.; Corney, D.; Papadopoulos, S.; Skraba, R.; Goker, A.; Kompatsiaris, I.; Jaimes, A.; (2013) "Sensing Trending Topics in Twitter" , *Multimedia, IEEE Transactions on* , vol.15, no.6, pp.1268,1282, doi: 10.1109/TMM.2013.2265080
2. Allan, James; Carbonell, Jaime; Doddington, George, Yamron, Jonathan ; Yang, Yiming; (1998). "Topic Detection and Tracking Pilot Study Final Report" In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop pp. 194-218
3. Becker, H.; Naaman, M. & Gravano, L.; (2011), Beyond trending topics: Real-world event identification on Twitter, *in* 'Fifth International AAAI Conference on Weblogs and Social Media' .
4. Cataldi, Mario; Di Caro, Luigi; Schifanella, Claudio; (2010)." Emerging topic detection on Twitter based on temporal and social terms evaluation." In Proceedings of the Tenth International Workshop on Multimedia Data Mining (MDMKDD '10). ACM, New York, NY, USA, , Article 4 , 10 pages.
5. Chenliang Li, Aixin Sun, and Anwitaman Datta; (2012). Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (CIKM '12). ACM, New York, NY, USA, 155-164. DOI=10.1145/2396761.2396785
6. CLUTO A Clustering ToolkitRelease 2.1.1 (2003) George Karypis University of Minnesota, Department of Computer Science
7. Dai, Xiang-Ying; Chen, Qing-Cai; Wang, Xiao-Long; Xu Jun; (2010) "Online topic detection and tracking of financial news based on hierarchical clustering," Machine Learning and Cybernetics (ICMLC), 2010 International Conference on , vol.6, no., pp.3341-3346, 11-14
8. Dai, Xiangying; Sun, Yunlian; (2010), "Event identification within news topics," Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on , vol., no., pp.498-502, 22-24

9. El-Beltagy, S.R.; Rafea, A.;(2008)" KP-Miner: A keyphrase extraction system for English and Arabic Documents", Information Systems
10. Eslam Elsawy, Moamen Mokhtar, and Walid Magdy; (2014.) "TweetMogaz v2: Identifying News Stories in Social Media". In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14). ACM, New York, NY, USA, 2042-2044. DOI=10.1145/2661829.2661843
11. Huafeng, Xie; Fang, Wu; Xuying ,Lu; (2011) "Study on the Optimization Design of the Subject Indexing Based on the Word-frequency Statistics" computer and information science, vol. 4 no.2
12. Huang, Zhen ; Cardenas ,Alfonso F.,(2009) "Extracting Hot Events from News Feeds, Visualization, and Insights"
13. Jain, S.; Pareek, J;(2009) "KeyPhrase Extraction Tool (KET) for Semantic Metadata Annotation of Learning Materials," International Conference on Signal Processing Systems , vol., no., pp.625-628, 15-17.
14. Kock, N; (2015), "One-tailed or two-tailed P values in PLS-SEM?" International Journal of e-Collaboration, 11(2), 1-7.
15. Lopez, C.; Prince, V.; Roche, M.;(2010)"Automatic titling of electronic documents with noun phrase extraction," Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of , vol., no., pp.168-171, 7-10 Dec.
16. Makkonen ,Juha ;(2009)"Semantic Classes in Topic Detection and Tracking" University of Helsinki Finland, Department of Computer Science, PhD Thesis, Series of Publications A, Report A-2009-8 Helsinki, 165 pages
17. Okamoto, Masayuki; Kikuchi, Masaaki; (2009), "Discovering Volatile Events in Your Neighborhood: Local-Area Topic Extraction from Blog Entries." In Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology (AIRS '09)Springer-Verlag, Berlin, Heidelberg, 181-192
18. Petrovic, S., Osborne, M., Mccreadie, R., Macdonald, C., and Ounis, I. ; (2013) "Can twitter replace newswire for breaking news? "In: ICWSM - 13, 8-10 Jul 2013, Boston, MA, USA.

19. Phuvipadawat, S.; Murata, T.;(2010) "Breaking News Detection and Tracking in Twitter," Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE/WIC/ACM International Conference on , vol.3, no., pp.120,123, doi: 10.1109/WI-IAT.2010.205
20. Piegorsch, Walter W., and A. John Bailer ;(2005).” Analyzing environmental data”. John Wiley & Sons, 2005.
21. Renee R. Ha, James C. Ha; (2011) “Integrative Statistics for the Social and Behavioral Sciences”, SAGE Publications, Inc
22. Rosa, H.; Batista, F.; Carvalho, J.P.; (2014) "Twitter Topic Fuzzy Fingerprints," Fuzzy Systems (FUZZ-IEEE), IEEE International Conference on , vol., no., pp.776,783, 6-11
23. JulyRuchi Parikh and Kamalakar Karlapalem; (2013) “ET: events from tweets”, In Proceedings of the 22nd international conference on World Wide Web companion (WWW '13 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 613-620.
24. Schlotzhauer, Sandra (2007). *Elementary Statistics Using JMP (SAS Press)*(PAP/CDR ed.). Cary, NC: SAS Institute. pp. 166–169. ISBN 1-599-94375-1.
25. Seo ,Young-Woo ; Sycara, Katia ; (2004) “Text Clustering for Topic Detection”, Carnegie Mellon University
26. Shoukry,Amira.; (2013) “Arabic sentence level sentiment analysis”, The American University in Cairo, Thesis presented to Dept. of Computer Science and Engineering.
27. Tomokiyo, Takashi;Hurst, Matthew. (2003). “A language model approach to keyphrase extraction”. In Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18 (MWE '03), Vol. 18. Association for Computational Linguistics, Stroudsburg, PA, USA, 33-40.
28. Twitter API documentation .(2015) (dev.twitter.com/rest/public)
29. Wang, Canhui; Zhang, Min; Ru, Liyun; Ma, Shaoping; (2008). "An Automatic Online News Topic Keyphrase Extraction System," Web Intelligence and Intelligent Agent Technology,2008. WI-IAT '08. IEEE/WIC/ACM International Conference on , vol.1, no., pp.214-219, 9-12.
30. Wang, XiaoLing; Mu, DeJun; Fang, Jun; (2008) "Improved Automatic Keyphrase Extraction by Using Semantic Information," Intelligent Computation Technology and

Automation (ICICTA), International Conference on , vol.1, no., pp.1061-1065, 20-22 Oct. 2008

31. Wartena, C.; Brussee, R.; (2008) "Topic Detection by Clustering Keywords," Database and Expert Systems Application, 2008. DEXA '08. 19th International Workshop on, vol., no., pp.54-58
32. Witten, Ian H.; Paynter, Gordon W.;Frank, Eibe; Gutwin, Carl; Nevill-Manning, Craig G. (1999). "KEA: Practical automatic keyphrase extraction". In Proceedings of the fourth ACM conference on Digital libraries (DL '99). ACM, New York, NY, USA, 254-255.
33. Xu ,Rui; Wunsch, Donald C. ;(2009) " Clustering" IEEE Press Series on Computational Intelligence A JOHN WILEY & SONS, INC., PUBLICATION
34. Xu, Yong-Dong; Quan, Guang-Ri; Xu, Zhi-Ming; Wang Ya-Dong; (2009) "Research on Text Hierarchical Topic Identification Algorithm Based on the Dynamic Diverse Thresholds Clustering," Asian Language Processing, 2009. IALP '09. International Conference on , vol., no., pp.206-210
35. Zhang ,Yan; Shi, Lei; Sun, Bai; Kong ,Liang; (2009) "Web Forum Sentiment Analysis Based on Topics," Computer and Information Technology, 2009. CIT '09. Ninth IEEE International Conference on , vol.2, no., pp.148-153
36. Zhao, Y., & Karypis, G.; (2001). "Criterion functions for document clustering: Experiments and analysis" (Vol. 1, p. 40). Technical report.
37. Zhao, Yanyan and Qin, Bing and Liu, Ting and Tang, Duyu, ; (2014)"Social sentiment sensor: a visualization system for topic detection and topic sentiment analysis on microblog" in journal of Multimedia Tools and Applications, by Springer US doi: 10.1007/s11042-014-2184-y
38. Zimmerman, Donald W. (1997). "A Note on Interpretation of the Paired-Samples tTest". Journal of Educational and Behavioral Statistics **22** (3): 349–360.doi:10.3102/10769986022003349. JSTOR 1165289.

Appendix A

A sample of the data of the baseline system after being preprocessed

Tweet	Annotated topic	Extracted topic
بالفيديو الرجاء يحقق فوزًا تاريخيا ويُسقِط الأهلي بهدف قاتل أول مبارياته	مباراة الاهلي و الرجاء	الاهلي الرجاء
محافظ مطروح يمنح لاعبي الرجاء مكافأة 5 آلاف جنيه لفوزهم النادي الأهلي	مباراة الاهلي و الرجاء	الاهلي الرجاء
حكم المباراة يعلن انتهاء مباراة الأهلي 1 2 الرجاء ويستهل الأهلي مبارياته الدوري بهزيمة	مباراة الاهلي و الرجاء	الاهلي الرجاء
الرجاء يفوز الأهلي بهدفين مقابل هدف أولى مباريات الأحمر بالدوري العام	مباراة الاهلي و الرجاء	الاهلي الرجاء
طارق سالم يحرز الهدف الثاني الرجاء تمريرة عمرو المنوفي ويتقدم الأهلي الدقيقة الثانية الوقت بدل الضائع	مباراة الاهلي و الرجاء	الاهلي الرجاء
هدف التعادل للأهلي الرجاء وقع متعب التسلل	مباراة الاهلي و الرجاء	الاهلي الرجاء
الرجاء يحافظ نظافة شباكه الأهلى مرور 60 دقيقة	مباراة الاهلي و الرجاء	الاهلي الرجاء
متعب يتعادل لـلأهلي عرضية صبري رحيل الدقيقة 79	مباراة الاهلي و الرجاء	الاهلي الرجاء
عمرو المنوفى يتقدم للرجاء الأهلى الدقيقة 75	مباراة الاهلي و الرجاء	الاهلي الرجاء
الرجاء يتقدم الأهلى بهدف نظيف الشوط التانى	مباراة الاهلي و الرجاء	الاهلي الرجاء
هدف تعادل رائع للأهلي برأسية عماد متعب الرجاء 1 1 الأهلي	مباراة الاهلي و الرجاء	الاهلي الرجاء
76 عمرو المنوفي يفاجئ الأهلي بالهدف الأول لصالح الرجاء الرجاء 1 0 الأهلي	مباراة الاهلي و الرجاء	الاهلي الرجاء
الأزهر الشريف ردًا خطط تركية لمنافسته حاول النيل منا فشل فشلًا ذريعًا	الازهر	الازهر
الإمام الأكبر إعفاء الطفلة رحمة المصروفات إتمام دراستها الأزهر	الازهر	الازهر
شيخ الأزهر يكلف عميد «الشريعة والقانون» برئاسة «الأزهر» خلفًا للعبد	الازهر	الازهر
اختيار عبد الحي عزب رئيسًا لجامعة الأزهر	الازهر	الازهر
أحمد الطيب يكلف عبد الحى عزب برئاسة جامعة_الأزهر	الازهر	الازهر
الأزهر ردا خطط تركية لمنافسته الفشل مصير حاول النيل منا	الازهر	الازهر
وفد الأزهر يزور الطفلة رحمة بمستشفى الشرطة بالعجوزة	الازهر	الازهر
شيخ الأزهر الشريف يدعو أبناء الوطن جميعا استلهام معاني التض	الازهر	الازهر
وزير الأوقاف يهنئ عزب برئاسة جامعة الأزهر	الازهر	الازهر
عباس شومان وكيل الأزهر تكليف عبدالحي عزب برئاسة جامعة الأزهر خلفا للدكتور أسامة العبد رئيس الجامعة وكان عزب يشغل عميد كلية الشريعة والقانون	الازهر	الازهر
شيخ الأزهر يكلف عميد «الشريعة والقانون» برئاسة «الأزهر» خلفًا للعبد	الازهر	الازهر

الازهر	الازهر	شيخ الأزهر يهنئ المصريين بذكرى انتصارات أكتوبر تقدم الدكتور أحمد الطيب شيخ الأزهر، بالتهنئة للشعب المصري، وال
الازهر	الازهر	عبد الحي عزب رئيسا لجامعة الأزهر
الحجاج	الحج	فيديو لاندلاع حريق بأحد مخيمات الحجاج عرفات
الحجاج	الحج	رئيس بعثة للطيران الحج انتهينا استعدادتنا لعودة الحجاج بدءًا الثلاثاء المقبل لنقل
الحج	الحج	الحج كان بياخد 3 شهور وساعات أكتر، لازم الناس تتاجر وهي رايحة وجاية يلاقوا ياكلوا يعملوا شوبنج Fahima75 seksek
الحج	الحج	وهل ترى بقه الواحد يلبّي وسط الشوبنج عادي؟ تخيّل كده ناس عمّالة تقيس هدوم ولبيك اللهم لبيك
الحج	الحج	رئيس بعثة حج القرعة صحة لاندلاع حرائق خيام عرفات بالسعودية
الحج	الحج	بعثة الحج تنفي نشوب حريق بمخيمات عرفات
حسام حسن مرتضى منصور	اقالة حسام حسن	مرتضى_منصور طلبت حسام_حسن عدم إشراك عبد_الشافي توقيع عقد إعارته لكنه أشركه بمباراة الداخلية وتسبب لنا إحراج
الزمالك	الزمالك	رسميًا تعيين محمد صلاح مدرب عام الزمالك تفاصيل أكثر
حسام حسن مرتضى منصور	اقالة حسام حسن	مرتضى_منصور اتفقنا جهاز حسام_حسن إبراهيم_حسن إخلاء الساحة لمدرب أجنبي المرشحين البرتغال فرنسا ألمان
الزمالك	الزمالك	مرتضى_منصور رئيس نادي الزمالك علاء عبد الغني يساعد محمد صلاح قيادة الزمالك بشكل مؤقت الكورة_في_الملعب
الزمالك	الزمالك	الكورة_في_الملعب الزمالك يعين محمد صلاح مدربًا للفريق لحين الاستقرار مدير فني أجنبي
حسام حسن مرتضى منصور	اقالة حسام حسن	الكورة_في_الملعب إقالة الجهاز الفني لنادي الزمالك بقيادة حسام_حسن

Appendix B

A sample of the Python code

```
def extract_unigrams (Tweets) :
    for tweet in Tweets:
       tokens=tweet.split()        //returns all unigrams of the tweet
       for token in tokens:
          list_of_unigrams .append(token)
    return list_of_unigrams
```

```
def average_freq (list_of_unigrams) :
    fd1=FreqDist()          //function used to calculate the frequency of each unigram
    //getting an descending order list of words based on their frequencies without duplication
    for unigran in list_of_unigrams:
            fd1.inc(unigram)
    //calculating the average frequency of unigrams with frequency>10
    for d in fd1.keys():
       if (int("".join(str(fd1[d])))>10):
          count=count+1
          nsum=nsum+int("".join(str(fd1[d])))
    avg=float(nsum)/float(count)
    return avg
```

```
def extract_significant_unigrams (list_of_unigrams, theta1 ):
        fd1=FreqDist()
        for unigram in list_of_unigrams:
                fd1.inc(unigram)
        for unigrm in fd1.keys():
           if (int("".join(str(fd1[unigrm])))>=theta1:    ##threshold
                    str_word1 = " ".join(unigrm)
                    str_word1 = unigrm[:(len(unigrm)-1)
                    significant_unigrams.append(str_word1)
        return significant_unigrams
```

```
def  extract_tweets (Tweets, significant _unigram):
        for tweet in Tweets:
                    tokens=tweet.split()
                    for token in tokens:
                       if (token == significant_unigram):
                          associated_tweets_set.append(token)
        return associated_tweets_set
```

```
def average_PF (associated_tweets_unigrams):
    fd2=FreqDist()
    for unigram in associated_tweets_unigrams:
        fd2.inc(unigram)

    pf1count=0

     count=0

    for word in fd2.keys():

        pf1count=pf1count+int("".join(str(fd2[word])))

        count=count+1

    avg_pf=pf1count/count

    return avg_pf
```

```
def extract_FCU(associated_tweets_unigrams, theta2 ):

    fd2=FreqDist()
    for unigram in associated_tweets_unigrams:
        fd2.inc(unigram)

    for unigrm in fd2.keys():

        if (int("".join(str(fd2[unigrn])))>=theta2:

            str_word1 = " ".join(unigrm)

            str_word1 = unigrm[:(len(unigrm))]

            if (str(fd2[unigrm]))>0 :

                FCU.append(unigrm)

    return FCU
```

```
def Add_tweet_to_topic (associated_tweets_set, topic_tweets):
    for tweet in associated_tweets_set :
        topic_tweets.append(tweet)

def similar (FCU1 , FCU2 , theta):
    common = [ ]
    flag = FALSE
    for word1 in FCU1 :
        for word2 in FCU2 :
            if (word1 == word2 ):
                common.append (word1)
    if ( len(common) >= (len (FCU1) + len (FCU2)) * theta) :
        flag = TRUE
    return flag
```

```
def Content_similarity (significant_unigrams, associated_tweets_set, FCU, theta3, theta4 ,
alpha) :
    keywords= { a }     //set of significant unigrams representing trending topics, initially
                          contains an arbitrary value
    t= 1     // index of number of trending topics
    for i in range  ( 1, len(significant_unigrams)) :
        topic=[ ]
        topic_tweets=[ ]
        if ( significant_unigrams [i] not in keywords) :
            topic.append( significant_unigrams[i])
            Add_tweet_to_topic( associated_tweets_set[i],topic_tweets)
            for j in range (i+1 , len ( significant_unigrams )) :
                if (similar ( FCU[i], FCU[j] , theta3):
                    topic .append( significant_unigrams[j])
                    keywords.append(significant_unigrams[j])
                    Add_tweet_to_topic
                    (associated_tweets_set[j],topic_tweets)
                    for k in range (j+1 , len (significant_unigrams )) :
                        if (similar ( FCU[j] , FCU[k] , theta4):
                            topic.append( significant_unigrams[k])
                            keywords.append(significant_unigrams[k])
                            Add_tweet_to_topic
                            (associated_tweets_set[k],topic_tweets)
        if  ( len ( topic_tweets >= alpha ) :
            print "topic " + " " + t + "\n"
            for word in topic:
                print word + " "
            for tweet in topic_tweets:
                print tweet + "\n"
            t=t+1
```

```
def Trend_Topic_Extraction (Tweets) :
    list _of_unigrams= [ ]
    list_of_unigrams = extract_unigrams (Tweets)      //extracting unigrams of all tweets in
                                                       the data set
     theta1 = average_freq (list_of_unigrams)
    significant_unigrams = [ ]
    significant_unigrams = extract_significant_unigrams (list_of_unigrams, theta1 )
    m=len(significant_unigrams)
    associated_tweets_set= [ ]
    associated_tweets_unigrams =[ ]
    FCU = [ ]

    for i in range (1,m) :
        associated_tweets_set[i] .append( extract_tweets (Tweets, significant
        _unigrams[i]))
        associated_tweets_unigrams[i] .append (extract_unigrams
        (associated_tweets_set[i]))
        theta2 = average_PF (associated_tweets_unigrams[i])
        FCU[i] .append (extract_FCU(associated_tweets_unigrams[i] , theta2 ))
    alpha=20
    Content_similarity (significant_unigrams, associated_tweets_set, FCU, theta3, theta4 ,
    alpha)
```
