Multiple Post Microblog Summarization

David Inouye
University of Colorado at Colorado Springs

***Abstract*—The use of microblogs such as Twitter[1] has increased incredibly over the past few years. Because of the public nature and sheer volume of text from these constantly changing microblogs, it is often difficult to fully understand what is being said about various topics. A method for summarizing popular topics of microblogs has been proposed but its summaries are only one sentence or phrase in length. Therefore, this work focuses on extending microblog summarization by producing multiple post summaries. Two main summarization algorithms are explored: a clustering based algorithm and a threshold based Hybrid TF-IDF algorithm. The results will be evaluated by comparing the generated summaries with manually generated summaries. For purposes of comparison, the results are also compared to MEAD, LexRank and TextRank—some leading traditional multi-document summarization systems.**

***Index Terms*—microblogs, summarization, clustering.**

I. Introduction

THE massive rise of microblogging as a new form of communication and data generation[2] has opened up a new area of natural language processing that could be aimed at discovering real time public opinion or news stories. In order to understand and use this amount of information, however, automatic summarization is a necessity. Though automatic summarization of longer or more structured documents has been researched [1]–[9], processing short and unstructured microblog posts has only recently been considered. Sharifi, Hutton and Kalita [10] proposed and implemented two methods of summarizing any given microblog topic in one sentence.[3] They implemented a novel graph-based Phrase Reinforcement algorithm and a Hybrid TF-IDF algorithm. The Hybrid TF-IDF algorithm was an altered TF-IDF algorithm (explained in [11], [12]) in which the TF (term frequency) component is computed upon the entire collection of posts whereas the IDF (inverse document frequency) component is computed upon a single post. Both algorithms produced human competitive results but the Hybrid TF-IDF algorithm seemed to produce consistently better results and therefore is integrated into this project.

II. Motivation

Though microblog summarization algorithms have produced summaries competitive with human generated summaries [10], they only produce a single sentence. Consequently, they can only represent one idea surrounding a topic. With this limited coverage of a specified microblog topic, important or interesting information about a topic may be easily overlooked. Though these short summaries may provide simple *indicative* summaries that give enough information to spark the interests of users as explained in [5], multiple post summaries that cover multiple subtopics of the original topic would push the summaries towards being *informative* [5]. Therefore, this paper describes some possible methods for producing these multiple post summaries for microblogs.

III. Problem Definition

The problem considered in this paper is how to produce a multiple post summary of microblog posts on a particular topic that is specified by a keyword or phrase. The resulting summary will be an extractive summary because the algorithms presented in this paper extract quotes from the original collection of posts.

This problem can be defined as follows: given a topic keyword or phrase T and the number of posts for the summary k, retrieve a set of microblog posts P in which for all $p_i \in P$, T is in the text of p_i and output a set of posts S with a cardinality of k in which all $s_i \in S$ are related to T by a relative relevancy ranking r_i but for all $s_i, s_j \in S, s_i \not\sim s_j$. For each post $s_i \in S$, a feature vector $v_i \in V$ can be computed based on word frequencies after any noise has been removed. The primary measure of similarity will be the cosine similarity measure:

$$\text{sim}(s_i, s_j) = \cos(v_i, v_j) = \frac{v_i^t v_j}{\|v_i\|\|v_j\|},$$

which can be simplified to $\text{sim}(s_i, s_j) = v_i^t v_j$ because $\|v_i\| = \|v_j\| = 1$.

IV. Proposed Solution

Two main methods for producing summaries are explored in this paper. The first method is using clustering to cluster posts into subtopics and then summarizing each cluster individually and the second method is modifying the Hybrid TF-IDF summarization algorithm so that it can produce multiple post summaries. Initial testing of microblog post clustering are performed to select the best clustering method for the cluster summarier.

A. Clustering Microblog Posts

In this phase, Sharifi's program filters the posts by removing any non-English posts and spam messages as determined by simple heuristics and a spam classifier. Then, post noise such

D. Inouye is participating in a Research Experience for Undergraduates (REU) with the Department of Computer Science, University of Colorado, Colorado Springs, GA, 80918 USA.

[1] http://twitter.com

[2] http://www.networkworld.com/news/2010/041410-biz-stone-says-twitter-has.html

[3] The program developed by [10] will be referred to as "Sharifi's program" throughout the rest of the paper.

as html tags, website addresses, headings and references are removed.

Once the posts have been pre-processed, the feature vectors $v_i \in V$ will be computed for each post based on the Hybrid TF-IDF weighting of words already implemented in Sharifi's program. In order to increase the performance of the algorithms, the feature vector computation ignores two types of noise: stop words that appear in a large majority of posts such as "a," "and" or "the" and simple sentences in a post such as "Wow!!!" and "Hahaha, that's funny." These feature vectors are normalized to unit vectors so that $v_i' = \frac{v_i}{||v_i||}$ in order to account for the difference in post lengths [13]. However, for comparision, the first set of tests does not normalize the feature vectors. The posts are then processed by a variety of greedy clustering algorithms. The main set of algorithms are variations of the k-means algorithm. Because the standard k-means algorithm generally performs well and is easy to implement [14], it has been tested first. The bisecting k-means algorithm was implemented afterwards because it may perform better than the direct k-means algorithm as suggested in [13], [14]. The k-means++ algorithm, which is a new variation of k-means algorithm proposed and initially tested by Arthur [15], was then implemented. Finally, an algorithm that combines the k-means++ algorithm with the bisecting algorithm was implemented.

For the following definitions, the centroid $c_i \in C$ is defined as

$$c_i = \frac{\sum_{v \in V_i} v}{n_i},$$

in which n_i is the number of posts and V_i is the set of all feature vectors in ith cluster. The k-means clustering algorithms are defined as follows:

1) Standard k-means algorithm
   a) Randomly choose k initial centroids $c_i \in V$ from all the computed feature vectors.
   b) Assign each post p_i to the centroid that is most similar to its corresponding feature vector v_i.
   c) Compute the centroid of each cluster.
   d) Repeat steps 1b and 1c until no posts are reassigned.
2) Bisecting k-means algorithm
   a) Split the set of posts P into 2 clusters using the standard k-means algorithm ($k' = 2$) defined by step 1 above.
   b) Choose an already formed cluster to split.
   c) Repeat steps 2a and 2b until the kth cluster has been formed.
3) k-means++ algorithm
   a′) Choose initial centroids based on probability.
      i) Choose an initial centroid c_1 uniformly at random from V.
      ii) Choose the next center c_i, selecting $c_i = v' \in V$ with the probability $\frac{D(v')^2}{\sum_{v \in V} D(v)^2}$ where $D(v)$ is the shortest distance from v to the closest center which is already known.
      iii) Repeat step 3a′ii until k initial centroids have been chosen.
   b-d) Continue with the standard k-means clustering algorithm defined in steps 1b-1d.
4) Bisecting k-means++ algorithm
   a) Follow step 2a of the bisecting algorithm above except use the k-means++ algorithm instead of using the standard k-means algorithm.
   b-c) Continue with the bisecting k-means clustering algorithm defined in steps 2b-2c.

B. Summarization

1) Baseline Summarizers:

a) Random Summarizer - For a baseline, a random summarizer was implemented that randomly chose four posts out of all the posts in each topic to serve as a summary.

b) Mead Summarizer - For the well-known well known multi-document summarization system called MEAD[4] described in [16], the summaries were summarized with the default settings.

c) LexRank Summarizer - LexRank [17] is a graph based multi document summarization method that uses the similarity between two sentences as the weight of the edge between those two sentences. Then, the final score of a sentence is computed based on the weights of the edges that are connected to it. Since the MEAD summarization toolkit came with a LexRank feature script, the LexRank implementation with the MEAD toolkit was used to compute the LexRank summaries.

d) TextRank Summarizer - TextRank [18] is another graph based method that comes primarily from the ideas behind the PageRank [19] algorithm. Because the exact implementation was not available, the summarizer was implemented internally using the formulas described in [18].

2) Cluster Summarizer: The cluster summarizer used the best clustering algorithm found in the clustering tests—the normalized bisecting k-means++ algorithm. It clustered the posts into 4 clusters (k = 4) and then summarized each cluster with the Hybrid TF-IDF algorithm.

3) Variable Cluster Summarizer: This test varied the value of k for k-way clustering from 5 to 10 in order to see if changing the number of clusters affected the results of the Cluster Summarizer. The largest 4 clusters were then chosen, and then each of these four clusters was summarized with the Hybrid TF-IDF algorithm.

4) Hybrid TF-IDF Summarizer: This algorithm developed by [10] weights all the sentences based on a modified TF-IDF (Term Frequency Inverse Document Frequency) weighting of sentences. The definition of what a document is for microblog posts needed to be modified. Therefore, a hybrid definition of a document was used instead in which the TF component uses the entire collection of posts as one document while the IDF component is computed on each post individually. Originally, the algorithm only selected the best summarizing topic sentence, but for this project, it was modified to select the top four most weighted posts.

[4]http://www.summarization.com/mead/

V. Experimental Setup

A. Evaluation Methods

In order to test the clustering algorithms, a testing corpus of pre-classified posts has been fed to the algorithm and the values of entropy and purity has been used as the primary metrics as defined by [13]. Given a particular cluster X_i of size n_r, the entropy of the cluster is defined as

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

where q is the number of classes in the pre-classified posts and n_r^i is the number of posts of the ith class that were assigned to the rth cluster. The total entropy of the clustering solution is

$$E(X) = \sum_{r=1}^{k}\frac{n_r}{n}E(X_r).$$

In general, the smaller the entropy values the better the clustering solution. Similarly, the purity of a particular cluster is defined as

$$P(X_r) = \frac{1}{n_r}\max(n_r^i),$$

which represents the fraction of the cluster that is made up of the largest class of documents. The total purity of the clustering solution is

$$P(X) = \sum_{r=1}^{k}\frac{n_r}{n}P(X_r).$$

In general, the larger the purity values, the better the clustering solution.

In order to test the final automatic summaries, the ROUGE-N metric [20] will be used to compare manually generated summaries to the automated summaries because [10] found that for microblogs the ROUGE-1 (N = 1 for unigrams) metric is a sufficient evaluation of microblog summaries. Given that M is the set of manual summaries and u is the set of unigrams in a particular manual summary, ROUGE-1 can be defined as

$$\text{ROUGE-1} = \frac{\sum_{m\in M}\sum_{u\in m}\text{match}(u)}{\sum_{m\in M}\sum_{u\in m}\text{count}(u)},$$

where count(u) is the number of unigrams in the manual summary and match(u) is the number of co-occurring unigrams between the manual and automated summaries. This formulation of the ROUGE-N metric can be used to measure the precision of the auto summaries since the divisor is the number of relevant unigrams. The metric can be altered slightly so that it measures the recall of the auto summaries by changing the divisor to be the number of unigrams in the auto summary which represents the number of retrieved unigrams. This formulation of the ROUGE metric can be stated as follows:

$$\text{ROUGE-1} = \frac{\sum_{m\in M}\sum_{u\in m}\text{match}(u)}{M_n * \sum_{u\in a}\text{count}(u)},$$

where M_n is the number of manual summaries and a is the auto summary. Because both recall and precision are important in summaries, the F_1-measure of the precision and recall are computed such that

$$F_\beta - \text{measure} = \frac{(\beta^2+1)pr}{\beta^2(p+r)},$$

where p is the precision and r is the recall. In addition, for reporting the results, the average F_1-measure of all the iterations—25 iterations for non-random summarizers and 2500 for random summarizers—was computed as

$$\text{avg}(F_1\text{-measure}) = \frac{1}{F_n}\sum_{f\in F} f,$$

where F is the set of all F_1-measures and F_n is the number of F_1-measures being averaged.

At least two volunteers will manually generate multiple post summaries by performing all the main steps of the algorithm so that the basic steps parallel the steps of the algorithm. First, like the algorithm, they will cluster the posts into a specified number of clusters k. The specific value of k = 4 was chosen after looking at the posts and determining that on average 4 clusters seemed to be reasonable.[5] Second, they will choose the most representative post from each cluster. And finally, they will order the posts in a way that they think is most logical. The information from this last step was not used in this research project but was collected to possibly help an extension of this project that could deal with the ordering of the posts and post order coherence.

B. Test Data

The test data used in this research project came from the test data collected for Sharifi's summarization program [10]. Over the course of five consecutive days, Sharifi et. al. collected 1500 microblog posts from the top ten trending topics on the Twitter home page. Then, because microblog posts are an unstructured and informal way of communicating, these post were preprocessed to remove spam and other noise features. These pre-processing steps were as follows [10]:

1) Convert any HTML-encoded characters into ASCII.
2) Convert any Unicode characters (e.g. "nu24ff") into their ASCII equivalents and remove.
3) Filter out any embedded URL's (e.g. "http://"), HTML (e.g. "<a.../a>"), headings (e.g. "NEWS:"), references (e.g. "[...]"), tags (e.g. "<...>"), and retweet phrases (e.g. "RT" and "@AccountName").
4) Discard the post if it spam.
5) Discard the post if it is not in English.
6) Discard the post if another post by the same user has already been acquired.
7) Reduce the remaining number of posts by choosing the first 100 posts.
8) Break the post into sentences.
9) Detect the longest sentence that contains the topic phrase.

[5]Though the choice of k will introduce some bias into the manual summary generation, a specific value needs to be set in order to accurately compare the automatic summaries–which take k as a parameter–to the manual summaries especially because the ROUGE-1 metric is very sensitive to summary length. A possible extension of this project would be to design an algorithm to compute the best value for k given a set of posts.

These pre-processing steps and their rationale are described more fully in [10]. Fifty topics of 100 posts gives a total of 5,000 posts. For the clustering tests, the topics were split into 10 sets of 5 so that 5-way clustering could be evaluated over 10 different data sets. For the summarizing tests, only the first 25 topics were used because of the limited number of volunteer hours that were needed to perform manual summaries of the topics.

C. Clustering Test

In order to avoid the sensitivity of random seeding, 100 5-way clustering solutions were computed for each of the 10 different data sets for a total of 1000 iterations per algorithm.

D. Summarization Tests

For the summarizers that involve random seeding (e.g. ClusterSummarizer, RandomSummarizer), 100 summaries were produced for each topic to avoid the effects of random seeding.

For the well-known multi-document summarization system called MEAD[6] described in [16], the posts were summarized with the default settings. Each post was formatted to be one document with a single sentence inside of it.

Since the MEAD summarization toolkit came with a LexRank feature script, the LexRank implementation with the MEAD toolkit was used to compute the LexRank summaries. One change from the main MEAD program test, however, is that all the posts for each topic were added to one document as separate sentences so that their LexRank scores could be computed against each other.

Because the exact implementation of TextRank was unavailable, the summarizer was implemented internally using the formulas described in [18].

The cluster summarizer used the best clustering algorithm found in the clustering tests—the normalized bisecting k-means++ algorithm. It clustered the posts into 4 clusters (k = 4) and then summarized each cluster with the Hybrid TF-IDF algorithm. Because some of the volunteers had suggested that some topics had significant noise and the fact that k-way clustering can perform significantly differently for different values of k, the number of clusters was varied from five to ten and the largest 4 clusters were chosen to summarize.

Because the Hybrid TF-IDF may produce very similar sentences as the top most weighted sentences, a similarity threshold was applied in which the algorithm looped through the posts starting at the most weighted and only choosing the post if the following condition was true for the current post s_i:

$$\text{sim}(s_i, s_j) \leq t$$

for all $s_j \in R$ where R is the set of posts aleady chosen and t is the similarity threshold. The cosine similarity measure was used and the threshold was varied from 0 to 0.99 with increments of 0.01 for a total of 100 tests.

[6]http://www.summarization.com/mead/

VI. Results and Evaluation

A. Clustering Results and Analysis

he entropy and purity measures of the 1000 iterations for each algorithm were averaged to give an overall sense of each algorithm's performance. The algorithms marked "modified" normalized the feature vectors to unit lengths. The average purities and entropies for all 8 implementations are shown in Table I. In order to give an overview of the relative performance of each implementation, Figure 1 shows the relative entropies and purities that have been normalized based on the following equations:

$$E_i'(X) = \frac{\max(E(X))}{E_i(X)} \text{ and } P_i'(X) = \frac{P_i(X)}{\min(P(X))}.$$

Because of this normalization, higher values are better for both entropy and purity, and all the algorithms are relative to the base k-means algorithm.

TABLE I
Average Entropy and Purities

Algorithm Implementation	Avg. Entropy	Avg. Purity
k-means	0.740	0.491
k-means++	0.731	0.499
Bisecting k-means	0.732	0.504
Bisecting k-means++	0.724	0.509
k-means (modified)	0.732	0.499
k-means++ (modified)	0.724	0.508
Bisecting k-means (modified)	0.720	0.514
Bisecting k-means++ (modified)	0.709	0.525

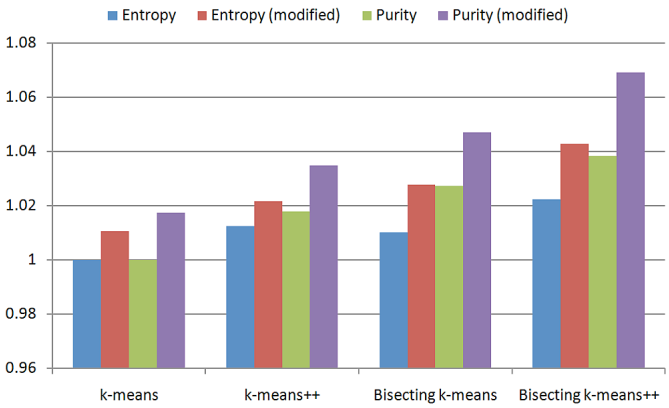

Fig. 1. The normalized relative entropies and purities for each algorithm. The series labeled "modified" use the normalized unit feature vector.

One of the first observations is that normalizing the feature vectors to be unit vectors improved the performance of all the algorithms by about 1.5%. This seems to be logical because normalizing the feature vectors significantly reduces the algorithms' sensitivity to length. The modified bisecting k-means++ algorithm performed the best by producing approximately 4% better entropy and 7% better purity than the base k-means algorithm. The bisecting k-means++ algorithm performed the best most likely because it combines the strengths of both the k-means++ algorithm and the bisecting

k-means algorithm. As suggested by [13], [14], the bisecting k-means algorithm did work better than the standard k-means algorithm. Though the k-means++ algorithm performed better than the standard k-means algorithm, the k-means++ algorithm did not perform as well as expected considering the tests performed in [15]. This may be due to the feature vector of short microblog posts.

Though these relative values show that the k-means algorithm can be improved, the absolute entropy and purity values as seen in Table I seem to suggest that the variations on the k-means algorithm will only produce small changes in the results. Hopefully, the different implementations of the criterion optimization algorithm will provide a significant increase in performance. In addition, the computation for the feature vectors may need to be reexamined because the clustering algorithms depend on good feature extraction.

B. Summarization Results and Analysis

1) Manual Summaries: Though the manual to manual ROUGE-1 F-measure scores seem to low (F-measure = 0.3291), this can be explained by the several factors. First, the instructions for summarizing did not give any guidelines to how each person clustered except for whatever themes or topics the volunteers thought could be good clusters. Therefore, the clusters for a topic may have been significantly different from one person to another depending on how they wanted to differentiate the posts. They were not limited to simply extractive clustering either since they were allowed to abstract concepts from the posts. In addition, for some topics, there was only thematic overlap rather than specific word overlap. For example, the topic "#MM" was a topic that stood for "Music Mondays" and the tweets would simply have names of songs or names of artists. Obviously, the names of songs or artists do not tend to overlap naturally. These results also seem to agree with the low F-measure scores that computed for one sentence summaries in Sharifi's work [10].

Because choosing the specific number of clusters for the volunteers (k = 4) could have introduced bias, data was collected on whether on not the volunteer thought 4 clusters was the right size for each particular post. This was a way to gather information about how good 4 clusters seemed and for future extensions of this research. Out of the 50 manual summaries—2 summaries per topic with 25 topics—the volunteers answered that there should have been less 13 times, the same 28 times, and more 9 times. This data is summarized in Figure 2.

It seems that the number of clusters (k = 4) was about the mean but was not always the best choice for all topics. Therefore, this research could be extended to discover how the number of k could be decided more intelligently.

2) Baseline Summarizers:

a) Random Summarizer - The seemingly high F-measure of the random summarizer may possibly be explained by a few characteristics of microblog posts. First, many microblog posts about a subject use the similar words in the post so unigram overlap within all posts seems to already be fairly high. Second, the most representative post about a topic is often quoted verbatim so the random summarizer has a decent chance of selecting one of these retweeted posts. Third, the unigram overlap even among manually generated summaries was low so it is not surprising that the random summarizer agreed some of the time.

b) MEAD - Interestingly, the MEAD default summarizer did slightly worse than the random summarizer. This seems to suggest that traditional ways of summarizing do not work very well with microblog posts. The unstructured and informal nature of the posts do not correlate with the expectations of the MEAD summarizer.

c) LexRank - Though the LexRank summarizer improved the random summarizer's F_1-measure by about 5%, it does not seem to be significantly better than the naïve random summarizer. Again, this seems to suggest that summarization of microblog posts is

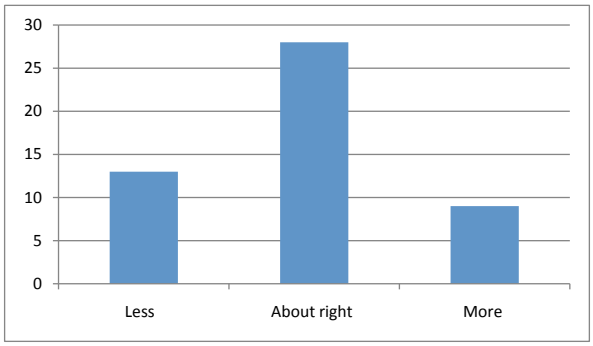

Fig. 2. Volunteer's answers to question about whether the specified number of clusters (k = 4) was right for the topic.

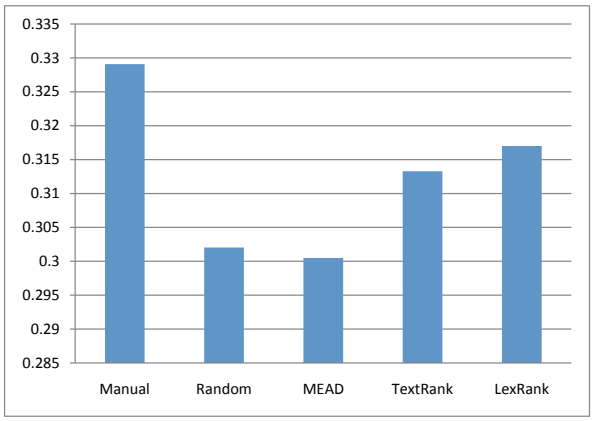

Fig. 3. F-measures for the baseline summarizers compared to the manual to manual F-measure.

significantly different than normal document summarization.

d) TextRank - Again, TextRank seemed to perform about 3.7% better than the random summarizer but not a significant improvement.

3) Cluster Summarizer: The cluster summarizer produced good results with an F-measure that is 8% better than the random summarizer. And though varying the number of computed clusters might have reduced noise, it seems from the results when $k > 4$ that the performance decreases as is shown in Figure 4. Therefore, the best Cluster summarizer is the original implementation that clustered the posts into 4 clusters and summarized each cluster into 4 representative posts.

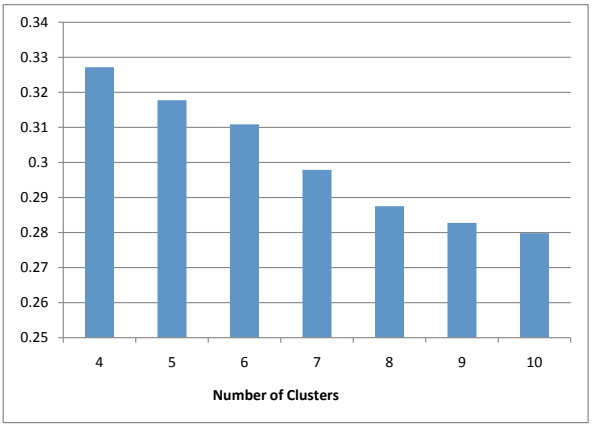

Fig. 4. F-measures of the Cluster Summarizer over the number of clusters.

4) Hybrid TF-IDF Summarizer with Similarity Threshold: The Hybrid TF-IDF Summarizer's performance was better than expected with a best F-measure of 0.3537 that was 17% better than the random summarizer when $t = 0.77$. Its best threshold measure of $t = 0.77$ seems to be reasonable because it allows for some overlap but does not allow sentences to be nearly identical. One reason for this summarizer doing so well is that it puts all the noise posts near the bottom since they do not seem to be related to other posts. In addition, the specific weighting of sentences is probably better suited for microblog posts than most traditional weightings of sentences such as normal TF-IDF.

5) Summary of Results: A summary of the best performing summarizers can be seen in Figure 6. The average precision, recall and F-measure are scaled by the F-measure of the random summarizer (0.3020) to give a relative sense of each summarizer. The values of precision, recall and F-measure with a standard deviation σ can be seen in Table II.

It can be seen from Figure 6 that the Hybrid TF-IDF and the Cluster summarizers performed better than any of the other summarizers including TextRank and LexRank. In addition, the Hybrid TF-IDF significantly improves over the Cluster summarizer by reaching about 18% better than random.

Because the topic phrase will be included in every post, it seems that a unigram match of the topic phrase is actually trivial and could be hiding non-trivial unigram overlap. Therefore, the ROUGE scores were recomputed so that the computation ignored keywords. The results are shown in Table III and summarized in Figure 7. Again, the results shown in Figure 7 are scaled by the F-measure of the random summarizer (0.2071).

The drop of almost all the averages by about 0.1 when

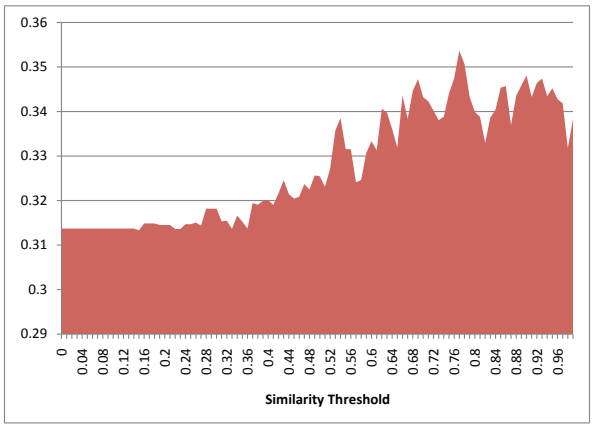

Fig. 5. F-measures of Hybrid TF-IDF Summarization algorithm over different thresholds.

TABLE II
ROUGE-1 AVERAGES

	Precision	Recall	F-measure $\pm$ σ
Manual	0.3383	0.3383	0.3291 $\pm$ 0.1089
Random	0.2885	0.3322	0.3020 $\pm$ 0.0930
Mead	0.2429	0.4109	0.3005 $\pm$ 0.0913
TextRank	0.2644	0.4075	0.3133 $\pm$ 0.0955
LexRank	0.3676	0.2943	0.3170 $\pm$ 0.0871
Cluster	0.3092	0.3606	0.3271 $\pm$ 0.1060
Hybrid TF-IDF	0.3505	0.3723	0.3537 $\pm$ 0.1172

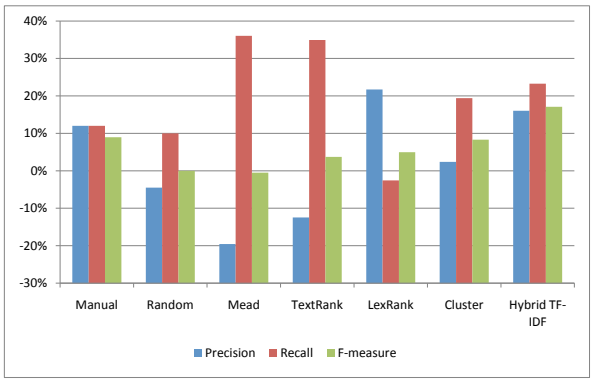

Fig. 6. Scaled F-measures of the summarizers.

keywords were ignored seems to be about right since the average length of a post is 11 words and each post will have one of the keywords at least once ($1/11 \approx 0.1$). However, the relative results of the summarizers changed. Using this slightly modified ROUGE metric, LexRank performs less than the random summarizer and the TextRank summarizer performs just slighltly better than the Cluster summarizer. The Hybrid TF-IDF summarizer continues to significantly outperform all other summarizers with an F-measure of 0.2524 which is 22% better than the random summarizer.

TABLE III
ROUGE-1 AVERAGES (KEYWORDS IGNORED)

	Precision	Recall	F-measure $\pm$ σ
Manual	0.2320	0.2320	0.2252 $\pm$ 0.0959
Random	0.1967	0.2283	0.2071 $\pm$ 0.0817
LexRank	0.2333	0.1894	0.2027 $\pm$ 0.0760
Mead	0.1771	0.3050	0.2204 $\pm$ 0.0738
Cluster	0.2180	0.2554	0.2310 $\pm$ 0.0891
TextRank	0.1954	0.3053	0.2328 $\pm$ 0.0799
Hybrid TF-IDF	0.2499	0.2666	0.2524 $\pm$ 0.0906

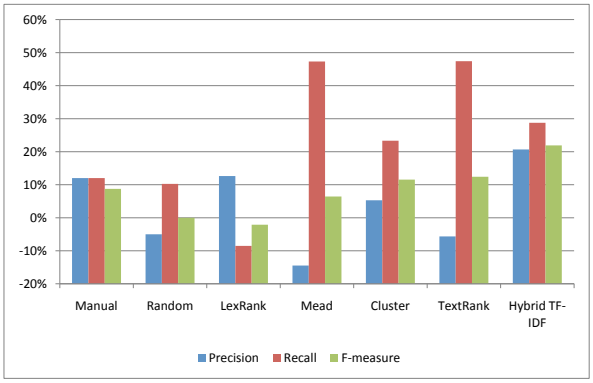

Fig. 7. Scaled modified F-measures (keywords ignored) of the summarizers.

Since the number of unigrams in the summary could affect the ROUGE scores, the average number of characters for each summarizer is shown in Figure 8. The high values of the TextRank and MEAD summarizer that are approximately 50% higher than the manual summaries, would explain why the recall values of the TextRank and MEAD summarizer are particularly high. In addition, the results help explain why the recall of every summarizer except the LexRank summarizer are higher than their corresponding precision measures. An extension of this work may be to attempt to penalize longer posts especially for the MEAD and TextRank summarizers to see if it improves their F-measures.

Examples of the top 3 summarizers (TextRank, Cluster and Hybrid TF-IDF) appear in Tables IV-VI. The three topics were chosen based on the F-measure scores of the Hybrid TF-IDF summarizer for it's best, worst and average topic.

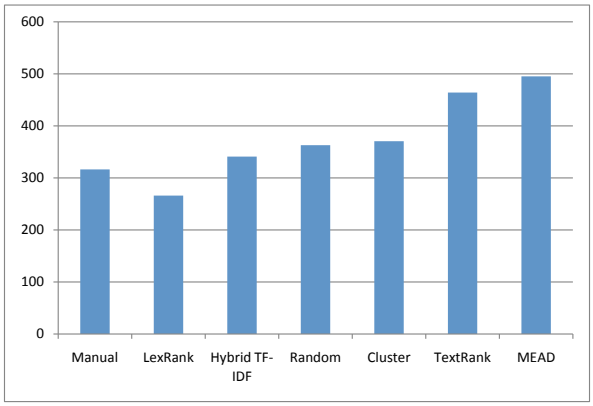

Fig. 8. Average number of characters per summarizer.

VII. CONCLUSION

The final goal of this research is to produce multiple post summaries of particular topics discussed in microblog posts in order to simplify and understand the information that microblogs provide. This research project specifically extends the work of [10], which only considered producing one sentence summaries. It seems from the clustering results that clustering microblog posts is not as simple or clean as clustering normal structured documents, and therefore, some new ways of clustering or computing feature vectors could be explored in future work.

The Hybrid TF-IDF summarizer with a similarity threshold of 0.77 produces significantly better results than the random summarizer and seems to be competitive with manually generated summaries. In addition, it performs better than some of the more traditional multi-document summarization systems MEAD, LexRank and TextRank. This points to the fact that microblog posts cannot be treated as traditional documents.

This project could be further extended in several ways. First, if a list of the most significant current topics could be computed as is being researched by [21], a summary of all the most significant topics could be generated in real time. It may also be possible to produce a topic browsing and summarization tool that will help people have a more comprehensive idea about real time microblog information.

Second, the coherency of the multiple post summary could be researched in depth. Sophisticated methods for ordering standard documents have been explored by [22], [23], and these advanced methods could possibly be applied to microblog summary cohesion. Other coherence issues such as pronoun resolution and fragmented arguments are issues that all summarization techniques need to consider [5]. If these issues could also be solved, clean, cohesive and comprehensive summaries of specified microblog topics could be practical and beneficial for many people.

TABLE IV
GOSSIP GIRL (BEST TOPIC FOR HYBRID TF-IDF)

Manual 1	about to watch gossip girl! great episode of Gossip Girl tonight! Not happy with that episode at all. #Gossip Girl gossip girl was way dramatic tonight blair and chuck can not break up
Manual 2	Yeah, it's time for Gossip Girl! great episode of Gossip Girl tonight! Gah, this week's Gossip Girl broke my heart in about 16 different ways. Chuck and Blair better make up soon! Just missed #gossip girl .....oh why? Oh, wait, i know.....too mauch damn homewrok!!
TextRank	i wanted to watch gossip girl with the girls but tummy ache :(this hasn't happened in months. ugh. house finally came on and it was gooood. So... I'm sitting in Indie's room with a bunch of Lesbians trying to explain the magnificence that is "Gossip Girl" Aww, Chuck & Blair are mad at each other. Usually Gossip Girl ends on a good note for Chuck and Blair. This episode didn't. I'm sad now :'(I will seriously stop watching "Gossip Girl" if Chuck and Blair break up. And WTF? Get Hilary Duff off of the show!
Cluster	has enjoyed throwing some MST3K style riffs at Gossip Girl tonight. Drew some inspiration from Bob Evil & Nick from Time Chasers. :D #fb gossip girl was way dramatic tonight blair and chuck can not break up Hmm. Gossip Girl voice overs starting to sound an awful lot like Meredith Grey's voice overs. This is not a positive development. wait people still watch gossip girl? lmao
Hybrid TF-IDF	Not happy with that episode at all. #Gossip Girl wait people still watch gossip girl? lmao great episode of Gossip Girl tonight! gossip girl was way dramatic tonight blair and chuck can not break up

TABLE V
#MM (WORST TOPIC FOR HYBRID TF-IDF)

Manual 1	its still monday?! welp #MM Tank "Slowly" -hmmmmm #MM "We Used To Vacation" Cold War Kids ¡– Pure Talent! Another one of my favs... #ralphlauren I got so many horses bitches call me polo... Guess the artist who said those lyrics #mm #musicmonday @Firefly2020 Thank you for #MM hugs - same back to you!
Manual 2	#MM The Feelies - The Good Earth #MM Keep It Flowin- Isley Brother...I could listen to this ALL day! Get up on it! 'Take your time when you likin a guy Cause if he sense that your feelings too intense, it's pimp or die..." #MM Jay-Z Soon You'll Understand iphone app which might mitigate this winter by starting the car from anywhere #iPhone #automobile #MM #app #apps http://bit.ly/6yZPE
TextRank	#MM Kanye West "See You In My Nightmare"... I Got The Right To Put Up A Fight!!! That is a sad discovery!! RT @joeyt2k just found out The Darkness broke up in 2006, is too devastated to speak. #MM #musicmonday #MusicMonday I always smoke dro, so it must be the answer, best beat in the game? my votes for #BeatCancer www.myspace.com/dezine420 #mm #MM #MusicMonday This is so very gay, but Miley Cyrus "Party in the U.S.A." is actually starting to grow on me...
Cluster	RT @aFOOLwperspctve: #MM Bobby Womack "If u think u lonely now" wheeeeeeeew my shit (Mine too!!) #MM Blade Icewood - Oh Boy, it's a detroit thing yall wouldnt understand lol #MM "Deosnt Mean Anything" by Alicia Keys...i love dis chick #MM Amy Winehouse Black to Black album...5stars
Hybrid TF-IDF	#MM Amy Winehouse Black to Black album...5stars @young_gab...stole my #MM song #MM "Deosnt Mean Anything" by Alicia Keys...i love dis chick RT @RoseGold88: #MM Rell feat jay Z-Love for free***thats my favorite song sis!!!

TABLE VI
A-ROD (AVERAGE TOPIC FOR HYBRID TF-IDF)

Manual 1	A-Rod and ARod are trending now...hahaha. Yankees fans: No matter how this postseason turns out, please shut up about A-Rod being a postseason choker. Yours in Christ, SDC Nice CC!!.. Posada's off his game tonight but A-Rod's on point! LET'S GO YANKEES! A-Rod is superman
Manual 2	A-Rod is just on fire this postseason. A-Rod homers in third straight game http://bit.ly/168LMB I HATE A - ROD TOO MUCH!!! WHO'S WITH ME? not only is ARod a trending topic but so is A-Rod lol
TextRank	Come on, Angels. Do work. Do something. Gosh, I haaaaaaate the Yankees. And A-rod is NOT worth that contract. Girardi is now going to pinch hit for A-Rod 3-2 here because that's what his book says is the right move (via @NoYoureATowel) Wow both A-Rod and ARod is on Trending topics. Stupidddd,,, Wow both A-Rod and ARod is on Trending topics. Stupidddd,,,
Cluster	I have no idea who none of these players are besides A Rod and Derek Jeter -_- A-Rod homers in third straight game: A-Rod homers in third straight game http://bit.ly/168LMB Gotta love that both arod and A-Rod are trending: Gotta love that both arod and A-Rod are trending LOL no one is in this game. Posada leaves home plate after Jeter's double play thinking it was 3 outs. kudos 2 A-rod who ran to cover home.
Hybrid TF-IDF	RT @johnnnyAa love this A-Rod guy, dude can really play baseball watching a-rod tie howard and gehrig's postseason rbi streak record. howard also tied gehrig's 70+ year old record just this year. Gotta love that both arod and A-Rod are trending: Gotta love that both arod and A-Rod are trending A-Rod homers in third straight game: A-Rod homers in third straight game http://bit.ly/168LMB

REFERENCES

[1] H. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[2] H. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.

[3] K. Mahesh, "Hypertext summary extraction for fast document browsing," in *Proceedings of the AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, 1997, pp. 95–103.

[4] K. Knight and D. Marcu, "Summarization beyond sentence extraction: a probabilistic approach to sentence compression," *Artif. Intell.*, vol. 139, no. 1, pp. 91–107, July 2002. [Online]. Available: http://dx.doi.org/10.1016/S0004-3702(02)00222-9

[5] U. Hahn and I. Mani, "The challenges of automatic summarization," *Computer*, pp. 29–36, 2000.

[6] N. Madnani, D. Zajic, B. Dorr, N. Ayan, and J. Lin, "Multiple alternative sentence compressions for automatic text summarization," in *Proceedings of the 2007 Document Understanding Conference (DUC-2007) at NLT/NAACL*. Citeseer, 2007, p. 26.

[7] E. Gonzàlez and M. Fuentes, "A New Lexical Chain Algorithm Used for Automatic Summarization," in *Proceeding of the 2009 conference on Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*. IOS Press, 2009, pp. 329–338.

[8] W. T. Visser and M. B. Wieling, "Sentence-based summarization of scientific documents the design and implementation of an online available automatic summarizer," 2008.

[9] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1995, pp. 68–73.

[10] B. Sharifi, M.-A. Hutton, and J. K. Kalita, "Automatic microblog classification and summarization," 2010.

[11] G. Salton, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988. [Online]. Available: http://dx.doi.org/10.1016/0306-4573(88)90021-0

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*, 1st ed. Prentice Hall, February 2000. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0130950696

[13] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.3151

[14] M. H. Dunham, *Data Mining: Introductory and Advanced Topics*. Prentice Hall, August 2002. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0130888923

[15] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: http://portal.acm.org/citation.cfm?id=1283494

[16] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang, "MEAD - a platform for multidocument multilingual text summarization," in *LREC 2004*, Lisbon, Portugal, May 2004.

[17] D. Radev and G. Erkan, "Lexrank: graph-based centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–480, 2004.

[18] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proceedings of EMNLP*. Barcelona: ACL, 2004, pp. 404–411.

[19] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine* 1," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[20] C. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, 2004, pp. 25–26.

[21] J. Benhardus, "Streaming trend detection in twitter," 2010.

[22] R. Barzilay, N. Elhadad, and K. McKeown, "Sentence ordering in multidocument summarization," in *Proceedings of the first international conference on Human language technology research*. Association for Computational Linguistics, 2001, p. 7.

[23] M. Lapata, "Probabilistic text structuring: Experiments with sentence ordering," in *Proceedings of the annual meeting of the Association for Computational Linguistics*, 2003, pp. 545–552.