RAProp: Ranking Tweets by Exploiting the Tweet/User/Web Ecosystem

by

Srijith Ravikumar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2013 by the
Graduate Supervisory Committee:

Subbarao Kambhampati, Chair
Hasan Davulcu
Huan Liu

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

The increasing popularity of Twitter renders improved trustworthiness and relevance assessment of tweets much more important for search. However, given the limitations on the size of tweets, it is hard to extract measures for ranking from the tweet's content alone. I propose a method of ranking tweets by generating a *reputation score* for each tweet that is based not just on content, but also additional information from the Twitter ecosystem that consists of users, tweets, and the web pages that tweets link to. This information is obtained by modeling the Twitter ecosystem as a three-layer graph. The reputation score is used to power two novel methods of ranking tweets by propagating the reputation over an *agreement graph* based on tweets' content similarity. Additionally, I show how the agreement graph helps counter tweet spam. An evaluation of my method on 16 million tweets from the TREC 2011 Microblog Dataset shows that it doubles the precision over baseline Twitter Search and achieves higher precision than current state of the art method. I present a detailed internal empirical evaluation of *RAProp* in comparison to several alternative approaches proposed by me, as well as external evaluation in comparison to the current state of the art method.

To my Parents.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to Subbarao Kambhampati for his advice and support. His excellent advising method combining guidance and intellectual freedom made my Masters experience both productive and exciting. It is constructive criticism that made me rethink my ideas and come up with better ideas.

I would like to thank to my other committee members Hasan Davulcu and Huan Liu for their guidance and support for my research and dissertation. Their exceptional co-operation and guidance significantly enabled a smoother dissertation process and contributed towards the higher quality of my research.

My big thanks to my collaborators Raju Balakrishan and Kartik Talamadupula for the contributions made and mentoring me. I also thank my Fellow Yochans – Sushovan De, Yuheng Hu, Preeth Inder Singh and William Cushing for the taking time out of their research to spend time discussing my ideas and providing me guidance when I needed it the most.

I am grateful to the Dr. Partha Dasgupta and Linda Chattin for their support as a teaching assistant through my studies. I would like to thank many other faculty in Arizona State University, without their courses and instruction this dissertation would not have been possible.

My special thanks to my father, mother and brother for encouragement and love throughout my life and education. Further, I would like to thank all my friends who made my graduate school a better experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Twitter, the popular microblogging service, is increasingly being looked upon as a source of the latest news and trends. The open nature of the platform, as well as the lack of restrictions on who can post information on it, leads to fast dissemination of all kinds of information on events ranging from breaking news to very niche occurrences. This has contributed even further to the growth of Twitter's user base, and has engendered the establishment of Twitter as a pre-eminent data source for users' queries – especially about hot topics – on the web. In a logical extension of this phenomenon, search engines and on-line retailers now consider real-time trends from tweets in their ranking of products, dissemination of news and in providing recommendations [2, 12] –leading to large-scale pecuniary implications. However, these monetary implications lead to increased incentives for abusing and circumventing the system, and this is manifested as microblog spamming. The open nature of Twitter proves to be a double-edged sword in such scenarios, and leaves them extremely vulnerable to the propagation of false information from profit-seeking and malicious users (*cf.* [23, 26]).

Spam techniques have proliferated much more widely on the open internet than they have on Twitter, so it should stand to reason that their effects can also be countered and negated to a large extent. Indeed, given powerful search techniques that can detect and neutralize spam-infused results, the average user should not even have to be aware of this issue. Unfortunately, Twitter's native search does not seem to consider the possibility of users crafting malicious tweets, and instead only considers the presence of query keywords, and the temporal proximity (recency) of, tweets ([28]). However, given Twitters model, an increase in the number of queries on a specific topic is usually also accompanied by an increase in the number of total tweets related to that topic. For example, when Apple releases a new model of the iPhone, Twitter searches related to this topic shoot up; however, so do the total number of tweets that contain the string iPhone.

To be precise, Twitter's native search does consider that the presence of a term in a large number of tweets is a strong indicator of popularity; indeed, it even measures such popularity for each tweet by measuring the number of re-tweets that it receives. While Twitter considers the number of re-tweet instances as the feature that make a particular tweet popular, it is not the sole such feature. In particular, this takes no note of *content-based similarity* and hence content-centric popularity; i.e., two tweets may not be related via a re-tweet, yet they may be semantically very similar. Another feature that is often muddled up due to re-tweets is *trust* – re-tweeting may not necessarily indicate trust, since most users re-tweet without verifying the contents of the tweet. Twitter tries to address this issue by filtering out spam tweets ([27]); however, while tweets that Twitter identifies as spam may well be untrustworthy, it cannot be assumed that tweets not marked as spam are all trustworthy. Moreover, providing correct and relevant information often requires more than just removal of spam (*cf.* [23]). Even when tweets are not malicious or deliberately manipulated, they may still be incorrect from a content perspective. Hence I need better ways to quantify and measure both the content-based popularity, as well as trustworthiness of a tweet given a query.

The function of a good ranking mechanism must be to rank tweets based on a combination of these features, such that the ones that are most relevant to a users query while being both popular and trustworthy float to the top, and malicious and less relevant tweets are suppressed to the lower portions of the results. Furthermore, since this ranking is an on-line operation that must displace the status quo, the computation time of the entire operation must also be acceptable. Such problems are relevant not just to Twitter itself, but also to search engines and on-line retailers that exploit Twitter trends for rankings and profit. Web search engines such as Google have dealt with similar problems that involve ranking web pages (documents) by using link analysis methods such as PageRank [6] to estimate the trustworthiness and popularity of pages. However, no such methods exist for tweets currently. Link analysis cannot be directly applied since there are no hyper links between tweets - the one link that does exist is re-tweeting, and the problems with using those exclusively have been outlined previously. Other approaches such as certifying tweets, or the users who post them, are both impractical and self-defeating. It is hard to verify millions of unknown existing users, and new users; besides, this leads to an evisceration of the very charm of open microblogging, which is that anyone may say anything. Furthermore, users who do not verify the veracity of information before re-tweeting – a phenomenon that is quite common, particularly for hot topics – contribute further to the propagation of false information that must be filtered by search methods.

Additionally, the single biggest difference between web and microblog search is the difference in the size of the documents – tweets are almost always much shorter. Finally, although trust assessment in the web and semantic web are addressed previously [17, 15, 25], these methods bootstrap from a seed set of nodes or edges with manually assigned trust values. However, large scale and real-time content on Twitter make such methods inapplicable for tweets.

1.1 My Approach

In response to the shortcomings of the current methods for ranking tweets, in this paper I propose *RAProp* (short for "reputation propagation"). My high level idea is similar to the TrustRank approach proposed by Gyöngyi et. al. ([17])–assess the reputation of the tweet messages, and propagate these scores over the inter-tweet endorsement structure. The realization of this idea is however complicated by two technical challenges: (i) it is not feasible to get reputation scores manually (as done by Gyöngyi et. al.) and (ii) there is no ready-made endorsement structure between tweets[1]. The main contribution of *RAProp* is effectively addressing these two challenges.

To avoid manual reputation assessment, I develop a method for automatically assessing reputation using the features of the Twitter eco-system consisting of tweet content, twitter users, linked web pages, and inter-relationships between these. To make-up for the lack of endorsement structure, I compute pairwise agreement between the content of individual tweets, and interpret the degree of agreement as the strength of endorsement among tweets.

More specifically, in *RAProp*, I model the twitter ecosystem as a three layer graph, consisting of a user layer, a tweet layer and a web page layer. The reputation of tweets are computed considering all three layers using a random-forest learner. Within the tweet layer, I exploit collective intelligence of the related tweets to derive the reputation, by considering re-tweets and the semantic similarity (*agreement*) between the tweets. Features from the user layer are transmitted to the tweet layer by using the "tweeted-by" links; features from the web layer are transmitted to the tweet layer by the URLs in the tweets. These features are then combined to find the initial reputation score. Final reputation score is computed by disseminating these scores among the tweet layer through the agreement and re-tweet links.

[1] Although re-tweeting can be viewed as endorsement, it provides a very sparse and biased endorsement structure.

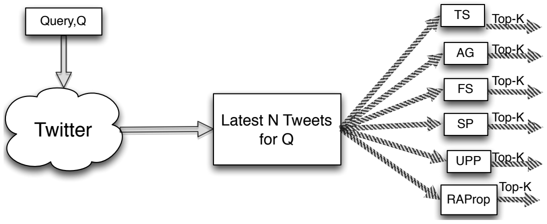


Figure 1.1: *The mediator model used, with the different methods I considered & my proposed method RAProp.*

I implement and evaluate *RAProp* within a mediator model, as shown in Figure 1.1. The implementation thus assumes access only to a ranked set of microblogs, rather than the entire tweet dataset.

It is thus useful for search engines that wish to exploit real-time results from microblogs to improve the results they return (e.g. news search). Many of these search engines do not own or house all of the relevant microblog data, but have ways of querying it at search time. Further the mediator assumption also makes the implementation applicable to any microblog service as long as it provides minimal query only access ([30]).

I present a detailed internal empirical evaluation of *RAProp* in comparison to several alternative approaches proposed by me, as well as external evaluation in comparison to the current state of the art method and Twitter Search. I also present the internal and external evaluation of my method in a non-mediator model where we assume to have all the tweets pre-indexed. My evaluation is done on the TREC microblog dataset (consisting of 16 million tweets), and the results show that *RAProp* leads to significant improvements in precision of the ranked results.

1.2 Organization of Thesis

In Chapter 2, I explain how I model the Twitter ecosystem in order to extract a measure of reputation for each tweet. I then describe in Chapter 3 the notion of agreement between tweets, and in Chapter 4, I present my ranking methods which use the reputation scores and an agreement graph generated via the methods in the preceding Chapter. I then discuss alternative approaches to ranking in Chapter 5. I then explain why I believe my method does work better than other baselines and in Chapter 6 presents my evaluation, and the various results that validate my hypotheses. I conclude with an overview of related work.

Chapter 2

MODELING THE TWITTER ECOSYSTEM

The trustworthiness of a tweet may be derived from the trustworthiness of the user who tweets it, the trustworthiness of the web URL mentioned in the tweet (if any) and the popularity of the tweet. I model the entire tweet ecosystem as a three layer graph as shown in Figure 2.1. Each layer in this model corresponds to one of the characteristics of a tweet mentioned above – the content, the user, and the links that are part of that tweet. The first layer I consider consists of the set U of all users u such that a tweet t_u by u is returned as part of the result set R for the query. Measuring trustworthiness of a user by itself is a heavy researched topic [8, 29]. Most of these research concentrates on judging the trustworthiness of existing users in twitter. Hence it has no or low trustworthiness predicting capabilities for a user who was not part of the data set which was used for computing the trustworthiness. Since the user base of twitter is growing exponentially, I believe that my user trustworthiness algorithm needs to predict the trustworthiness of not just users that were in my dataset but also users that were not part of my dataset. Hence, I compute the trustworthiness of a user from the user profile information. The user features that I use are the follower count, friends count, whether that user (profile) is verified, the time since the profile was created, and the total number of statuses (tweets) posted by that user. Another advantage of computing trustworthiness of a user from the user profile features is that I would be able to more quickly adjust my trustworthiness score of a user in accordance with any changes that happens in the profile(e.g.. sudden increase in the number of followers).

The second layer consists of the content of the tweets in R; i.e., the tweets themselves. I select some features of a tweet that were found to do well in determining the

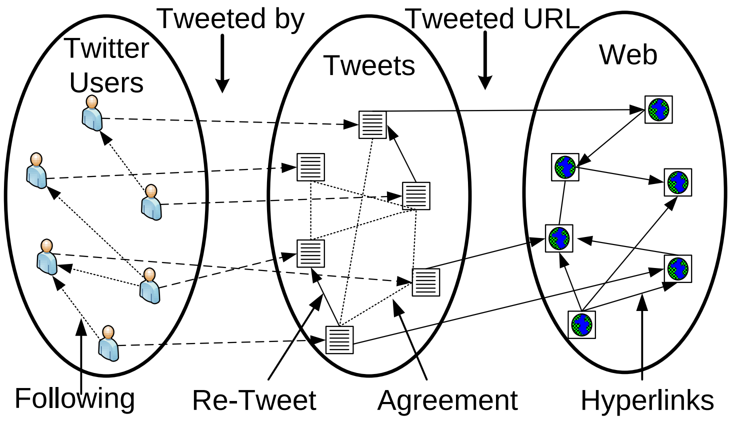


Figure 2.1: *Three layer ecosystem of Twitter space composed of user layer, tweets layer and the web layer.*

trustworthiness of that tweet [7]. The features I pick include whether the tweet is a re-tweet; the number of hash-tags; the length of the tweet; whether tweet mentions a user; the number of favourites received; the number of re-tweets received; and whether the tweet contains a question mark, exclamation mark, smile or frown. I believe that these features are a good indicator of the trustworthiness and relevance of content of the tweet. For example the presence of a smiley or a question mark in the tweet is a good indicator the tweet is not an authoritative account on that query topic. Hence the user may not be interested in such a tweet for that query and there by making it an indicator of relevance as well. To these features, I add a feature of my own: TF-IDF similarity which is weighed by proximity of the query keywords in the tweet. These features helps me determine the relevance of the tweet to the query. Due to the fact that tweets are short length documents, I do not expect the query terms to be present in the tweet multiple times. This leads to the Term Frequency (TF) being either 0 or 1. Proximity of the query keywords in the tweet is a very important feature when judging the relevance. This is because I cannot rely on the mere existence of the query keywords; most tweets returned by the Twitter search interface already contain all the keywords in the query. I try to account for this in my TF-IDF similarity score by exponentially decaying the TF-IDF similarity based on the proximity of the query terms in the tweet. Thus the similarity of a tweet r to the query Q is defined as:

$$S = \mathrm{T(t_i,Q)} \times e^{\frac{-w \times d}{l}}$$

where $T(t_i,Q)$ is the TF-IDF similarity of the tweet t_i to the query,Q, $w = 0.2$ is a constant that decides the weight for proximity score, l is number of terms in the query and d is the sum of distances between each term in the query to its nearest neighbour.

The third and final *web* layer consists of the links that are used in tweets. A number of tweets link to external websites, and it would be remiss to throw that information away when considering the relevance of tweets. Additionally, the web has an existing, easily query-able repository that scores web pages based on some notion of trust and influence – PageRank. For each tweet that contains a web link, I instantiate a node that represents that link in the web layer of the graph. There are links from that tweet to the node in the web layer, as well as intra-layer links among the nodes in the web layer based on link relationships on the open web.

The proposed ranking is performed in the tweets layer, but all three layers are used to compute what I call the *Reputation Score*. The agreement from the tweet layer, as well as the influence scores from the user layer and the trust/influence scores from the web layer are assigned to the respective tweets using the *inter-layer* links. In this way, each tweet collects its score, and at the end of this computation the entire set of returned results (tweets) R consists of scored tweets. I compute the Reputation Score for each tweet $r \in R$ using these features defined.

2.1 Computing Reputation Score

In order to rank the result set R_Q considering trustworthiness, I need a measure of trustworthiness for each tweet. However, for deriving these metrics there are no established or authoritative sources to compare against. Furthermore, as noted previously, tweets differ from web pages in that they are constrained to be much shorter and contain no explicit hyper links between them. This rules out the direct application of existing web search methodologies for trust assessment—like Pagerank [6]—or methods used in the Semantic Web [15]. The paucity of content means that additional *meta* information contained in each tweet may be mined and used in the search process. I use the user, web page and the tweet meta information as the features that determines the trustworthiness of a tweet.

To learn the reputation score from features, I use a Random Forest based learning to rank method.I train the Random Forest with the User, Tweet and Web features described previously. I used the gold standard relevance values (described in Chapter 6.1) for training and testing my model. I randomly pick 5% of the gold standard dataset for training the model, and another 5% to test the trained model (the remaining data is reserved for the experiments). Since I did not want to penalize tweets that do not contain a URL or user information that I were not able to crawl, I impute the missing respective feature values with population average. I normalize the reputation score to lie between 0 and 1. The time taken for the prediction of scores for each tweet using Random Forest is nearly negligible, which helps me keep my algorithm computation time to the minimum. Using the features chosen by this method, I get a score – the Reputation Score – for each tweet.

Chapter 3

AGREEMENT

Reputation of tweet is heavily influenced by the user who tweeted the tweet and page rank of the web page mentioned in the tweet. Hence, Reputation Score of a tweet may be considered to be more of a measure of trustworthiness or popularity of the user rather than relevance or popularity of the content of the tweet. For example, for the query *BBC News Staff Cuts*, the Result Set, R may contain tweets from BBC News that has no relevance for the query - *"BBC News: Police in bid to cut wages bill http://bbc.in/emdnQY"*. Although it is evident that the tweet has no relevance to the query, such tweets would gain high Reputation score as the user - *BBC News* - and the url - *http://bbc.in* - are reputed. I hypothesise that relevant tweets are those tweets that contain a popular and trustworthy content. As Popularity of a tweet is measured by the number of re-tweets it gets, the popularity of a content may be measured by the number of independent trustworthy users who endorse that content.

Approaches such as TrustRank [17] use the hyper-links among the web pages as a sort of endorsement structure to propagate trust on the surface web. Although the re-tweet relations among Twitter messages can be seen as a sort of endorsement, they fall far short both because of their sparsity and because re-tweeting is not nearly as deliberate as inserting hyper-links. In this chapter, I develop of a complementary endorsement structure among tweets by interpreting mutual agreement between two tweets as an implicit endorsement.

3.1 Agreement as a Metric for Popularity & Trust

Given the scale of data on Twitter, especially for popular topics, it is quite normal for a large number of tweets to refer to the same semantic concept or topic. The very notion of using re-tweets as a metric of popularity is based on the idea that a tweet which is tweeted out many times (re-tweeted) must be popular. Using agreement as a metric to measure popularity of a topic can be seen as a logical extension of using re-tweets to measure the popularity of a tweet. The very notion of using re-tweets as a metric of popularity is based on the idea that a tweet which is tweeted out many times (re-tweeted) must be popular. It is then a logical extension to consider not just the idea of *re-tweets as endorsements*, but also the presence of tweets that display a high degree of similarity with popular tweets (without being exactly the same) as independent verification. This kind of high degree of similarity can be computed from the pair-wise agreement of the content of two tweets, and this gives me a good way to measure the popularity of a tweet in terms of the number of other tweets that seem to be close to it.

Using agreement to measure the trustworthiness have been found to perform well [5]. If two independent users agree on the same fact – that is, they tweet out the same thing – it is likely that those tweets are trustworthy. As the number of users who tweet semantically similar tweets increases, so does the belief in the idea that those tweets are all trustworthy. Notice that this may not be the case with *syntactic* similarity (i.e., native re-tweets): in those cases, it is totally possible that an untrustworthy tweet is picked up and re-tweeted a number of times. However, tweets that are syntactically slightly different, but semantically very close (in terms of their content and topic) are often indicators of independent sources and thus trustworthiness. More generally, this idea is based off of the (web) notion that agreement is more indicative of trustworthiness than just native re-tweets. The reader is directed to work by Balakrishnan & Kambhampati [5] for a more general explanation of why agreement is likely to indicate trustworthiness and relevance. Therefore tweets that are syntactically slightly different, but semantically very close (in terms of their content and topic) are often indicators of independent sources and thus trustworthiness.

3.2 Agreement Computation

Computing the semantic agreement (as outlined above) between tweets at query-time while still satisfying timing and efficiency concerns is a challenging task. Due to this, only computationally simple methods may be realistically used. TF-IDF similarity has been found to perform well when measuring semantic similarity for named entity matching[10] and for computing semantic similarity between web database entities [5]. In the web scenario, IDF makes sure that more common words such as verbs are weighted lesser than nouns which are less frequent. But due to the sparsity of verb and other stop words in tweets, I noticed that IDF for some verbs to be much higher than the nouns and adverbs. Hence, I weight the TF-IDF similarity for each part of speech differently such that I weigh the tags that are important for agreement higher than other tags which does not highly correlate to agreement. I use a Twitter POS tagger [14] to identify the parts of speech of each tweet. Hence Agreement of a pair of tweet T_1,T_2 is defined as:

$$AG(T_1,T_2) = \sum_{t \in (T_1 \cap T_2)} TF_t(T_1) \times TF_t(T_2) \times IDF(t)^2 \times POSWT(t)$$

where $POSWT(t_i)$ is given as according to the table 3.1

I compute TF-IDF similarity on the stop word removed and stemmed result set, R. However, due to the way Twitter's native search (and hence my method, which tries to improve it) is set up, every single result $r \in R$ contains the query term(s) in Q. Thus the actual content that is used for the agreement computation - and thus ranking - is actually the *residual content* of a tweet. The residual content is that part of a tweet $r \in R$ which does not contain the query Q; that is, $r \setminus Q$. This ensures that the IDF value of the query

POS Tag	**POSWT()**
Proper noun	4.0
Common noun	3.0
pronoun	1.0
verb	1.0
adjective / adverb	3.0
interjection / preposition	.5
existential	.2
Hashtags	6.0
URL	8.0
Numberical	2.0

Table 3.1: Part of speech weights for various POS tags

term as well as other common words that are not stop words is negligible in the similarity computation, and guarantees that the agreement computation is not affected by this. Instead of normalizing the TF-IDF similarity by the normalization factor, I divide the TF-IDF similarity only by the highest TF value. Normalization was a necessity in web where web pages has no bound limit and normalization helps me penalize documents with large number of terms along with the query terms and give higher score to documents that have only fewer terms. But in the case of twitter, the document size is bound (140 characters), hence I do not penalize for using the entire 140 characters as they might bring in more content relevant to the query. I penalize tweets that repeat the terms multiple times as existence of the same term that agree on multiple times does not increase the agreement value.

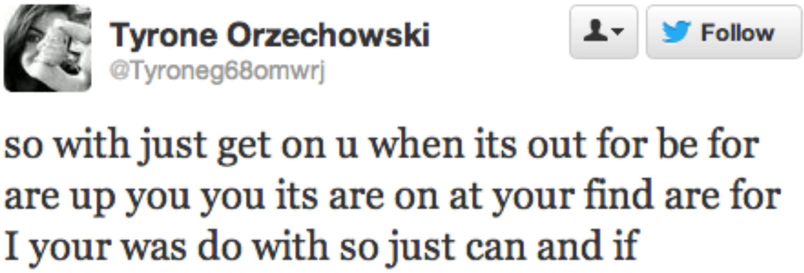
Chapter 4

RANKING USING AGREEMENT AND REPUTATION

The ultimate aim of any efficient search procedure is to rank the set R of returned results with respect to a given query Q in terms of the following: (1) relevance, and thus pertinence[1] of a specific result $r \in R_Q$ to Q; and (2) the trust reposed in r. These two (at times orthogonal) metrics must be combined into a single *score* for each r, in order to make the ranking process easier. Ranking based solely on the Reputation Score may not capture the true relevance of a tweet to the query, and ranking purely on agreement may lead to spam clusters gaining prominence.

A better approach is to propagate the Reputation Scores over the Agreement Graph; this can be seen as a trust-informed relevance assessment. I explain the construction of the Agreement Graph, and the propagation of the Reputation Score over it, in Chapter 4.2. I then explain why I restrict ourselves to a single propagation on that graph.

4.1 Picking the Result Set, R



so with just get on u when its out for be for are up you you its are on at your find are for I your was do with so just can and if

Figure 4.1: Example of a tweet that use excessive stop words

I do my experiments on two different methods. One where I assume a mediator model and other I assume I host the entire twitter data set. In the mediator model I assume

[1] A result r is pertinent to a query Q iff it contains at least one of the keywords $q \in Q$.

I don't host the data. For each query,Q' in the dataset, I collect the top-K results returned by twitter. These results becomes my initial result set,R'.

When I assume to host the entire data set, I do not need to simulate twitter. I query the data set my self and the top-K results,R' are picked from the list sorted by TF-IDF similarity of the tweet to the query.

The initial data set, R' is then filtered to remove any Re-tweets or Replies. I remove the re-tweets and replies from my results set as the gold standard considers these tweets as irrelevant to the query. As my method does not differentiate re-tweets and replies I remove these tweets as a prepossessing step.

I add more terms to the query,Q' to get the expanded query,Q. I select the expansion terms from the initial data set,R'. I pick the top-5 nouns that have the highest TF-IDF score. In order to constrain the expansion only to nouns, I run a twitter NLP parser [14] to Part of speech tag the tweets. The TFs of each noun is then multiplied with its IDF value to compute the TF-IDF score. The top-5 terms according to the TF-IDF score is added to the query. The top-N tweets returned by Twitter for the expanded query becomes the result set,R.

I believe that all words in the query term are not equally important. For example, stop words or verbs are much less important than a noun. On web documents, IDF usually takes care of weighting the nouns higher than stop words. But in the Twitter scenario, I noticed that IDF is high for even stop words due to sparsity of presence of stop words in tweets. This is especially important in the case of Twitter as there contains spam tweets that use just stop words as in Figure 4.1. These tweets try to match the stop words in the query in order to be part the results. Hence, I compute the TF-IDF similarity of result set,R

by weighting the Nouns 10 times higher than other word similarity. I also remove tweets that contain less 4 terms in them as these tweets mostly only contain the query terms and no other information.

Twitter matches query terms in URL as well while returning results. Thus, I add the URL as chunks split by special characters as part of the tweet in order for agreement to account for keywords present in the URL alone. The tweets are stripped out of punctuation,determiner,coordinating conjunction so that agreement is only over the important terms.

4.2 Agreement Graph

I believe that although the scores using the features do represent relevance and trustworthiness, there could exist tweets having a low Reputation score but still be relevant to the query and trustworthy. This is due to the fact that the Reputation Scores may be biased towards user popularity and the web page credibility and is not dependant on the actual content of the tweet. I want my ranking to be based on the actual tweet content and not based on how popular the user who tweeted and/or the web-page mentioned in the tweet are. However, finding trustworthy and relevant tweet at query time is a computationally expensive operation. Also, due to the reason that news are tweeted in real-time, there might not be any external source to verify the trustworthiness of the tweets. Hence I need to rely on user and web-page trustworthiness to determine if the content of the tweet is trustworthy.

Agreement is different from relevance to query; computation of pairwise agreement between any two tweets represents the similarity of their content to each other, not to the query Q. Also, tweets which have low relevance to the query term may form cliques between them and thereby gain high agreement. This problem is well known in other fields as well, for example PageRank [4] on the web.

Hence, I am not able to exploit Agreement or Reputation Score by itself to compute a trustworthy and relevant Result Set. But if I base my final ranking on the Reputation Score, I need to provide the tweets of unpopular users but trustworthy content with a higher Reputation score that they deserve. For the same, I use the agreement between the tweet as a measure of deserved Reputation Score of the tweet.I propagate the Reputation Score to the tweets that are trustworthy but are from less reputed users. The Reputation Score propagation may be seen as a method to find which out of each agreement clusters are more trustworthy. The more trustworthy cluster is expected to contain more number of reputed seed tweets. In the propagation step these seed tweets propagate their reputation to the tweets that not highly reputed but has high agreement with the reputed tweets.

My result set R_Q (for a specific query Q) is constructed such that all the tweets $t \in R_Q$ already bear some relevance to Q – tweets are chosen for inclusion in R_Q as they contain one or more keywords from the query, Q. I propagated the reputation on the agreement graph that is formed by the agreement analysis detailed above. This ensures that if there is a tweet in R_Q that is highly relevant to Q, it will not be suppressed simply because it did not carry enough reputation. More formally, I claim that the reputation of a tweet $t \in R_Q$ will be the sum of its current reputation and the reputation of all tweets that agree with t weighted by the magnitude of their agreement, i.e.

$$S'(Q,t_i) = S(Q,t_i) + \sum_{j \in E} w_{ij} \times S(Q,t_j) \; \forall \; (i,j) \in E$$

where w_j is the agreement between tweet t_i and t_j and E is the edges in agreement graph. The result set R_Q is ranked by the newly computed $S'(Q,t)$. In order to perform this computation, I create a graph such that the vertices represents the tweets and edges between the vertices representing the agreement between them. The tweets are ranked based on the reputation score computed after the propagation. I evaluate the performance of my method, *RAProp*, in my experiments in Chapter 6 and shows that it performs better than baseline.

Chapter 5

DISCUSSION OF DESIGN CHOICES

In the previous chapters, I focused on a specific approach, *RAProp*–that involved computing reputation scores using the features from the 3-layer Twitter eco-system, and propagating the reputation scores over the implicit inter-tweet endorsement structure in terms of the agreement graph. My framework however is general enough that it allows other variations for ranking tweets. In the following, I describe some of the more compelling variations and discuss their relative trade-off with respect to *RAProp*. In the next chapter, I will present empirical comparison of these variations to *RAProp*.

5.1 Ranking Just by Features (FS)

Ranking tweets based on only features has been attempted before [13, 19]. I compare the performance this kind of method – Feature Score (*FS*) – in my evaluations. Such methods make the assumption that all *reputed* tweets that are pertinent to the query are relevant as well. This is not always true - the reputation score may not capture the true relevance of a tweet to the query. For example, for the query "apple jobs", the top results as ranked by reputation may be about the Apple founder, Steve Jobs. However, the query may concern a recent jobs report that mentions Apple Computer Inc. In such cases, my approach, which uses the Agreement Graph created using the *residual content* of the tweets, is able to capture the popularity of the topic and therefore rank tweets pertaining to the more popular topic higher than a less relevant tweet with a higher reputation. My results show that my method indeed does perform better than using just the feature scores.

5.2 Ranking Just by Agreement (AG)

Another approach to ranking is by ranking tweets by considering only the agreement – using a *voting* methodology – where each tweet contributes to the other tweets' trust and hence ranking. This is used in the context of web sources by Balakrishnan et al. ([5]). However, the pairwise agreement between tweets represents the similarity of their content to *each other*, and says nothing about the relevance of the tweets to the query Q. This may lead to the formation of cliques of high agreement but low relevance within the result set, a problem that besets other voting methods. Agreement-based ranking is thus highly susceptible to irrelevant or untrustworthy tweet clusters occupying the top slots in the ranking. My experiments confirm this, as the agreement (*AG*) ranking, when used alone, has lower precision compared to my method.

5.3 Ranking with the Reputations of a Small Seed Set (SA)

I could also try to apply the ideas of Trust Rank [17] to the tweet scenario, where a small set of tweets called the *Seed Set* is picked and considered trustworthy. The reputation of that set is then propagated to all the tweets that agree with the tweets in the set. I pick the top-S tweets based on the reputation score from the result set R_Q, and denote this the seed set. The tweets are then finally ranked based on the Seed Propagated reputation score. I call this method *SP*, and see that my method outperforms even this method. Here I must note that restricting the seed set's size leads to the issue of picking the optimal size seed set, as well as the issue of picking seeds that are diverse enough with respect to the overall set R_Q. Since I compute the reputation score without the need for human experts, I am able to increase the size of my seed set without incurring significant additional overhead. I hypothesized that increasing the size of the seed set gradually to the size of the set R_Q would lead to an increase in relevance. This is borne out by my experiments. In the limiting case, I am essentially propagating the reputation score of every tweet in R_Q to all the tweets that agree with it – this turns into my novel ranking method, *RAProp*.

5.4 Ranking with Reputation Propagation to Fix-point

The number of propagations of reputation over the agreement graph may also be varied. Many other approaches that deal with the semantic web and the web consider the transitive nature of trust by using a multi-step propagation [3]. For Twitter, however, multiple propagations may perform poorer than just a single propagation step.

Consider the agreement graph which has tweets as the vertices in the graph and edges between them represents the agreement between the tweets. Each tweet has a Reputation Score associated with it. Unlike other approaches of propagation of scores over graph [17, 6], I do not propagate my feature scores over the seed agreement graph until reaching a steady state. I propagate the feature score over the seed agreement graph only once.

Unlike the web scenario, the links between tweets in my case are implicit links based on agreement. Thus, for a spam tweet to get agreement with a trustworthy tweet, all it needs to do is to agree with the content of the trustworthy tweet. This is not the case in web scenario where the trustworthy user is the one who controls the explicit links in that page. Let me illustrate this with an example of exploiting multiple propagation of Reputation score to rank a spam tweet higher in the ranked result.

Consider the scenario where a query on twitter, gives me the results such that there are two set of tweets, one set contains all the tweets that contain the content which is trustworthy and the other set contains all the tweets that are spam in nature. Since twitter does not consider trustworthiness of the tweets while ranking the tweets, it does not differentiate between either of the set. The tweets that contain the query terms are returned by twitter ranked by the chronological order. On the other hand in my case, I create the agreement

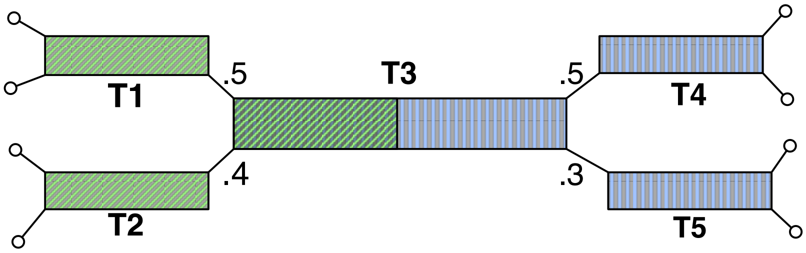
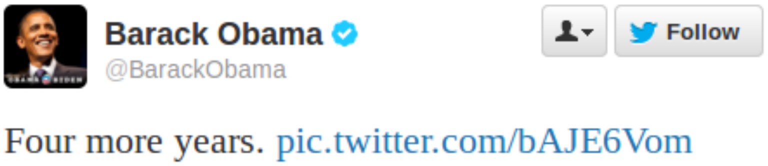


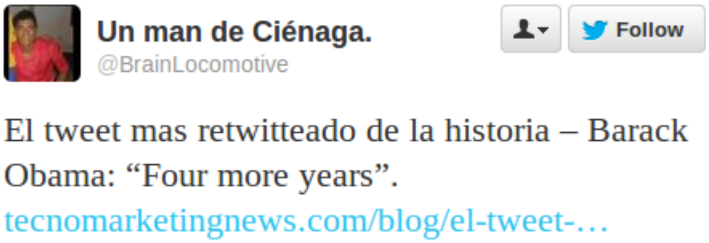
Figure 5.1: Illustration of propagation from trustworthy tweets to untrustworthy tweets



Barack Obama
@BarackObama

Four more years. pic.twitter.com/bAJE6Vom

Figure 5.2: A trustworthy tweet with trustworthy URL



Un man de Ciénaga.
@BrainLocomotive

El tweet mas retwitteado de la historia – Barack Obama: “Four more years”. tecnomarketingnews.com/blog/el-tweet-…

Figure 5.3: Spam tweet with trustworthy tweet text and spam URL

graph which would create two closely connected graph that are minimally connected. Let me assume the two graphs are connected by a spam tweet that tries to be part of the top results by quoting the most popular tweet of the trustworthy tweet and using rest of the tweet to input untrustworthy content as Figure 5.1. If I assume the case that I do multiple propagation over the seed agreement graph, the feature score from the trustworthy tweets (T1,T2) is propagated to the untrustworthy tweets (T4,T5) through the spam tweet, T3. Since I assume on seed score to represent the trustworthiness and popularity of the content, multiple iterations of the propagation would lead to untrustworthy tweets to be considered as trustworthy and be ranked higher in the results than they should be.

Let me illustrate the above scenario with a real example from twitter. Figure 5.2 shows the tweet by Barack Obama after he won the 2012 Elections. The tweet became so popular that it became the highest re-tweeted tweet. Spammer on seeing the popularity of the tweet and the content in the tweet, tried to capitalize on the same by trying to use the same content of the popular tweet and adding malicious content along with the same as in figure 5.3. This malicious tweet may be considered as the tweet that could propagate the trustworthiness from the trustworthy tweets to the untrustworthy tweets.

As the feature score s(Q,T) for each tweet is a measure of the trustworthiness and popularity of the tweet and the user who tweeted it, I expect T1,T2 to have higher feature score than T3,T4,T5. Hence I need to ensure that the during the propagation of the feature scores I do not propagate the feature scores from trustworthy tweets to untrustworthy tweets. Since I propagate the feature scores only once over the agreement graph, the untrustworthy tweets get only agreement from the other untrustworthy tweets eg. T4, T5; the trustworthy tweets gets agreement from other trustworthy tweets eg. T1,T2. As I assumed trustworthy tweets to have higher feature scores, they are expected to get higher scores after propagation and since untrustworthy tweets are expected to have low scores even after the propagation. The tweet that bridges the trust and untrustworthy graphs T3, gets agreement from trustworthy and untrustworthy tweets in accordance to its agreement with the untrustworthy and trustworthy tweets. Hence T3 is expected to get a feature score in between the trustworthy and the untrustworthy tweets.

Since I propagate the feature score s(Q,T) over the agreement graph just once, T3 gets the feature score from T1,T2 and T4,T5 but the score is not passed over to T4,T5. As I believe T1,T2 to have higher feature score than T4,T5 I would expect T1,T2 to be ranked above T4,T5 and T3 to be ranked in between.

Additionally, agreement values are noisy due to inherent errors in computing semantic agreement from text. Consequently, multiple propagations have a multiplying effect and may result in untrustworthy tweets gaining in the final ranking. I demonstrate this phenomenon in my experiments, by computing the precision values for multiple propagation steps.

5.5 Computing Agreement considering Similar words

Latent semantic analysis (LSI) [11] is an option that could be attempted in order to better compute agreement between tweets. LSI helps me to capture that some terms are closely related to other terms. These terms may be synonyms or terms that frequently co-occur. Hence Agreement over the tweets in the Latent Dimension helps me capture this similarity. But during my experiments I noticed that LSI over the Result Set, R seems to produce a large number of sparse dimension. Reducing the dimensions to an acceptable number of dimension as 30 gave me high level of loss of more than 60%. My experiments on the reduced dimension showed that it had degraded performance than plain TF-IDF similarity. I believe that the low document size of tweets does not help LSI capture synonyms and co-occurrence as effectively as it could in the case of web pages. Also, LSI computation took more than 30 minutes over a Result Set of size 2000.

5.6 Ranking using User,Page Rank Propagation(UPP)

Another idea of propagation may be to propagate just the trustworthiness using the agreement graph. Pagerank is a measure of trustworthiness of a web page; likewise the user features such as number of followers determines the trustworthiness of a user. PageRank may be considered as an external measure of the trustworthiness of the content i.e. when a user cites an external source such as *CNN* to substantiate his story, the story becomes more credible as *CNN* is a trusted source. Similarly a user with high trustworthiness is more likely to tweet trustworthy tweets. A tweet that has high agreement with a trustworthy tweet is also considered to be trustworthy.

The inter-tweet agreement is used to propagate the web page trust and the user trust to tweets that contain the same content. The trustworthiness acquired from propagated user and page rank scores is used as a features along with the tweet features(which determines the relevance of the tweet to the query) mentioned in Chapter 2 to rank the tweet according to relevance and trust. The features are input to the Random Forest Learning to Rank method. I believe ranking by propagation of user and page rank trust would do well on result set,R that contain agreement clusters with atleast one reputed user and/or a reputed web page linked. But on queries where there are clusters of reputed user and web pages and other clusters with no reputed user or web page, the propagation step does not propagate any trust. And there by degrading the method to accessing the relevance of a tweet based on just the tweet features. My experiments confirm this hypothesis that UPP would not perform well on all queries and there by having lesser average precision than my method.

Chapter 6

EVALUTION AND DISCUSSION

In this chapter, I present an empirical evaluation of my proposed approach *RAProp*. I do this by comparing it both to the Twitter's native search, as well as the variations I discussed in Chapter 5. I start by describing the dataset used for my experiments in Chapter 6.1. I then discuss my experimental set-up in Chapter 6.2, and then present results that demonstrate the merits of my approach in Chapter 6.3.

6.1 Dataset

For the purposes of this evaluation, the gold standard G_Q is a set of tweets annotated by TREC, where the annotations are with respect to their relevance to a given query Q. The relevance of each tweet is denoted by 3 discrete, mutually exclusive values $\{-1,\ 0,\ 1\}$: -1 stands for an untrustworthy tweet, 0 signifies irrelevance, and 1 stands for tweets that are relevant to the query. The gold standard gives me a way of evaluating tweets t in the search results. It is generated by humans who examine the relevance of tweets to given queries. Tweets containing the query keywords from Q are randomly sampled from a universal space of tweets U, and ranked according to the scheme mentioned previously.

For my evaluation, I used the TREC 2011 Microblog Dataset ([24]).This collection includes about 16 million tweets sampled from Twitter over a 2 week time period. It represents over 5 million microbloggers, at an average of 3 tweets per user. My experiments were conducted on the 49 queries that are provided along with this dataset (and thus 49 different gold standards, one for each query, as defined previously). The tweets in the gold standard provided by the TREC is generated by randomly sampling tweets containing the query keywords. Each tweet picked is given a relevance score of 0 and 1, 0 signifying the tweet is not relevant to the query and 1 signifying the tweet is relevant to the query. Of the 60129 tweets in the gold standard across 49 queries, 57048 are marked as not relevant to the query, 2081 are marked as relevant to the query and 116 are marked as spam. This means that even though there are 2081 tweets that are marked as relevant, there is no guarantee that a *given* query will feature k tweets that are highly relevant to it (and must therefore be ranked the highest by any good top-k ranking). Furthermore, whether such tweets are available for ranking also depends on the size of the set N from which tweets are chosen for ranking in the first place. All of these issues lead to lower precision rates when averaged across all queries. However, since the main purpose of my evaluation is to show my superiority other baseline approaches, this is not a major concern. The user information for all tweets in the dataset was crawled and stored off-line. I used the Pagerank API in order to collect the PageRank of all the web URLs mentioned in the tweets in this set.

6.2 Experimental Setup

Using the set of returned tweets R_Q that corresponds to a query Q, I evaluate each of the ranking methods, as shown in Figure 1.1. Since my dataset is off-line (due to the use of the TREC dataset and the gold standard as described above), I have no direct way of running a Twitter search over that dataset. I thus simulate Twitter search (*TS*) on my dataset by sorting a copy of R_Q in reverse chronological order (i.e., latest first). I also use the methods discussed in Chapter 5, as well as my proposed *RAProp* method, to rank R_Q. I set the bag size for my learning to rank method –Random Forest – as 10 and maximum number of leafs for each tree as 20 to avoid over-fitting to the training data.

6.3 Internal Evaluation of methods

I compare my method,*RAProp* against the other design choices mentioned in Chapter 5. I compare the precision of the different methods both while considering a mediator model as well as a non-mediator model. In the mediator model, I pick the top-N tweets that

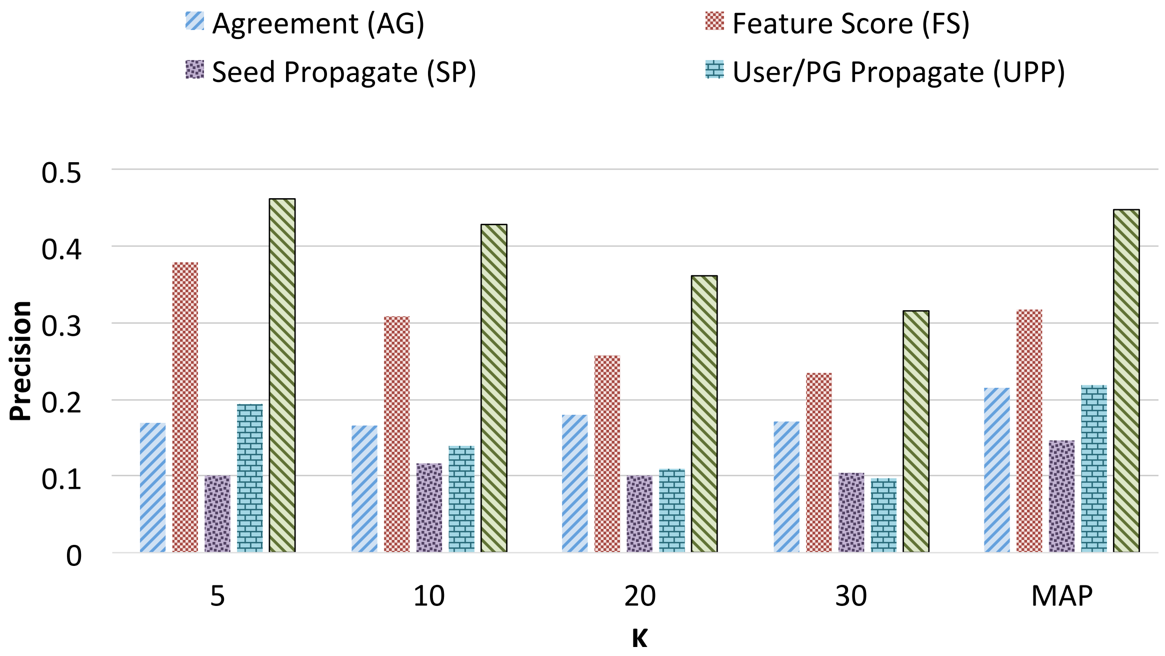


Figure 6.1: *Comparison of the proposed approach against other design choices in mediator model*

my simulated twitter returns and this is the input to all the various methods. In the non-mediator model, the top-N tweets is selected by the TF-IDF similarity of the tweet to the query.

6.3.1 Internal Evaluation of methods in a mediator model

I compared the top-K Precision at 5, 10, 20, 30 and MAP, of my method *RAProp* along with the various approaches proposed in Chapter 6.2. Since not all relevant tweets from the dataset for the query were part of the gold standard, I ignore those tweets that are not part of the gold standard while computing the precision value. I pick the N most recent tweets that contain one or more of the query keywords. For my experiments I set the seed size s for Seed Propagate (SP) to be 5 and $N = 2000$.

Figure 6.1, proves my hypothesis that *RAProp* has better precision values than using Reputation Score alone (*FS*) or Agreement (*AG*) alone for ranking. It also proves that I am able to achieve better precision scores by not restricting the propagation to a seed set as in Seed Propagation(*SP*). I also proves that propagation of trust alone and then using that as a feature for ranking, User/PG Propagate (*UPP)*,does not perform as well as my method. Since there exist less than *K* relevant documents in the Result Set *R*, the precision values are expected to drop as the value of *k* increases. However, *RAProp* maintains its dominance over the other methods and the baseline. Additionally, the MAP values show that *RAProp* is able to place relevant results higher as compared to the other methods.

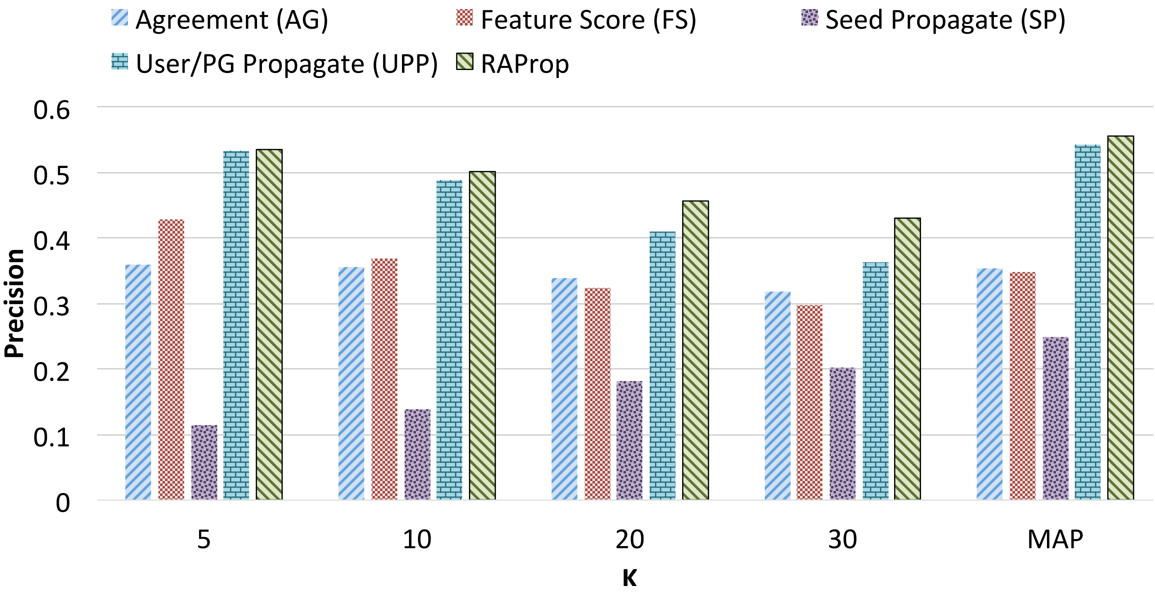


Figure 6.2: *Comparison of the proposed approach against other design choices in a non-mediator model*

6.3.2 Internal Evaluation of methods in a non-mediator model

I compared the top-*K* Precision at 5, 10, 20, 30 and MAP, of the proposed method assuming I have the entire twitter dataset. This allows me to choose the Result Set,*R* from the entire data set instead of top-*N* from simulated twitter results. I choose the Result Set,*R* by picking the top-*N* tweets according to TF-IDF similarity of the tweet to the query, as mentioned in Chapter 4.1.

As I can see from the Figure 6.2, my method gets better precision scores than all other design choices considered. This proves that my method, *RAProp* is able to achieve higher precision even on a non-mediator model where the Result Set,*R* is expected to have higher number of relevant documents. I also notice that although User/PG Propagate (*UPP*) has nearly comparable precision values as my method, it does not achieve higher precision scores than my method. Since, *UPP* achieves lesser precision scores and is computationally more expensive I believe that *RAProp* is a significantly better approach.

6.3.3 Varying the Size of Seed Set

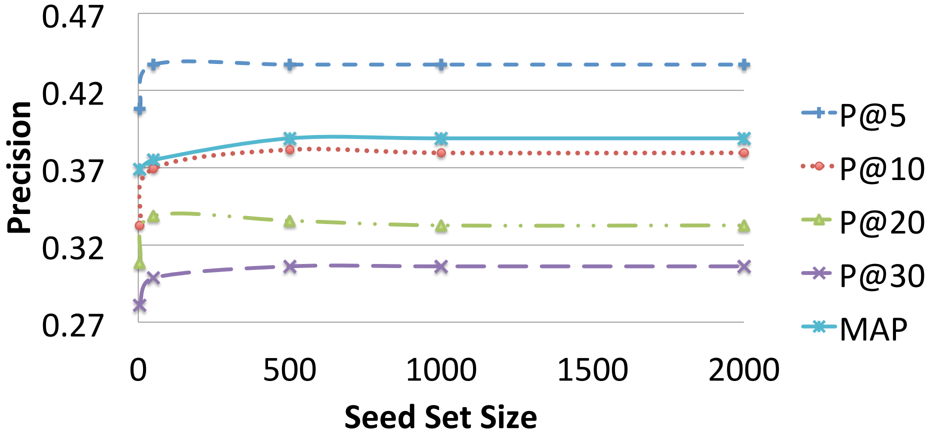


Figure 6.3: *Precision and MAP across various seed set sizes.*

As mentioned in Chapter 5, I varied the size of seed set to measure the change in precision values at each size of the seed set. I believe that the precision values level off at a seed set size lesser than *N*. I verify my hypothesis by computing the precision and MAP values at various values of the seed set size *S*; the results are as shown in Figure 6.3. As expected, there is an increase in the Precision and MAP values as I increase the seed set size, until a level off at size 500. Thus at some point, adding more tweets into the seed set fails to increase the Precision; however, I have no method of computing that exact cut-off point without an empirical evaluation, since it is likely to vary by dataset and query. However, given that I need to pre-compute the reputation scores of all tweets in order to get a seed set of tweets with the highest reputations (whether its size be 5 or 500), there is no significant additional overhead to increasing the seed set size gradually to *N*.

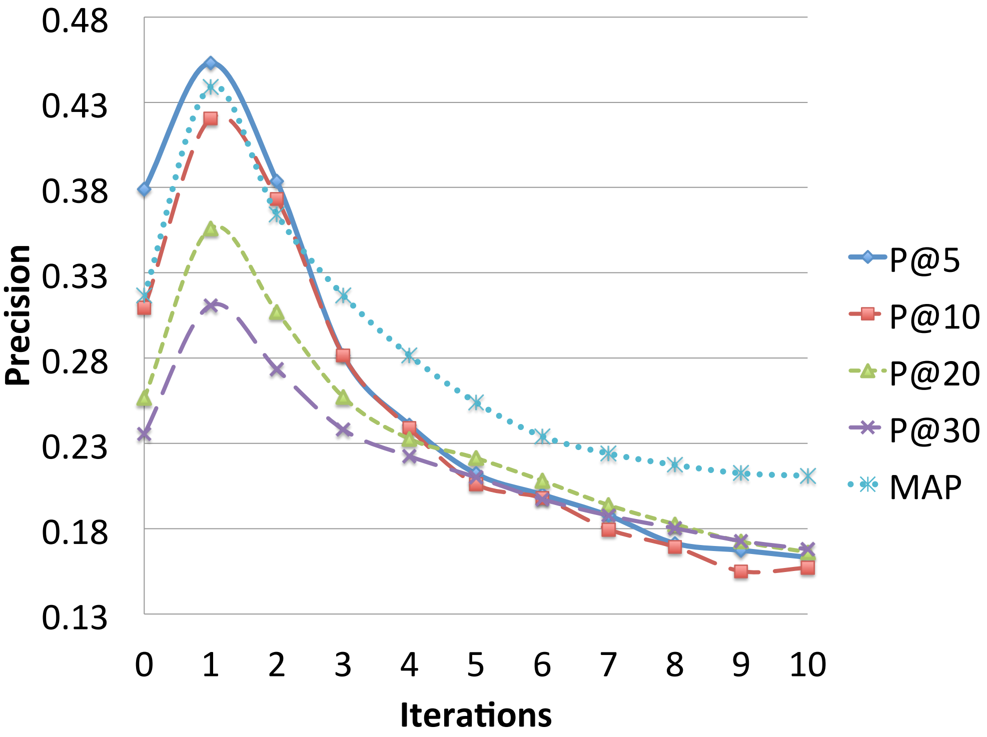


Figure 6.4: *Precision and MAP across multiple propagations of RAProp*

6.3.4 1-ply vs. *Multiple Ply*

I compare the top-*k* Precision at 5, 10, 20, 30 results and MAP (Mean Average Precision) values for various numbers of propagations over the Agreement Graph. Zero iterations can be considered as ranking based only on initial reputation scores, which is the *FS* method. One iteration over the agreement graph is the *RAProp* method. As shown in Figure 6.4, propagating the reputation score over the agreement graph certainly improves the Precision and MAP scores. However, I see that multiple iterations lead to a reduction of Precision and MAP scores. This validates my claim in Chapter 5 that multiple propagations will lead to a decrease in relevance.

6.4 External Evaluation of methods

In this section, I evaluate the performance of my method *RAProp* to two other external baselines, Twitter Search and USC/ISI method. In order to compare my method to the current Twitter Search(*TS*), I simulate Twitter Search over my data set by sorting the tweets that contain the query keywords in the reverse chronological order. I also compare my method with the TREC Microblog 2011 best performing method by Metzler and Cai (USC/ISI) [21]. USC/ISI uses a full dependence Markov Random Field Model, Indri, to achieve a relevance score for each tweet in the dataset. Indri creates a off-line index on the entire tweets dataset in order to provide a relevance score for each tweet in the entire tweets dataset. This score along with other tweet specific features such as length of tweet, existence of a URL or a hash-tag is used by a Learning to Rank method to rank the tweets. In our experiments, we compare the performance of our method against the USC/ISI method both as a mediator and non mediator. In the non-mediator model, we run the queries over the entire tweet dataset index. On the mediator model, since I assume I do not have access to the entire dataset. I create a per query index on the top-*N* tweets returned by twitter for that query as we assume that is the entire tweets available for that query.

I compare the performance of my method over these baselines while assuming a mediator model as well non-mediator model. As shown in Figure 6.5, when I assume a mediator model, my model *RAProp* achieves more than 200% higher precision than the default twitter search. Also, I am able to achieve higher precision than the current state of

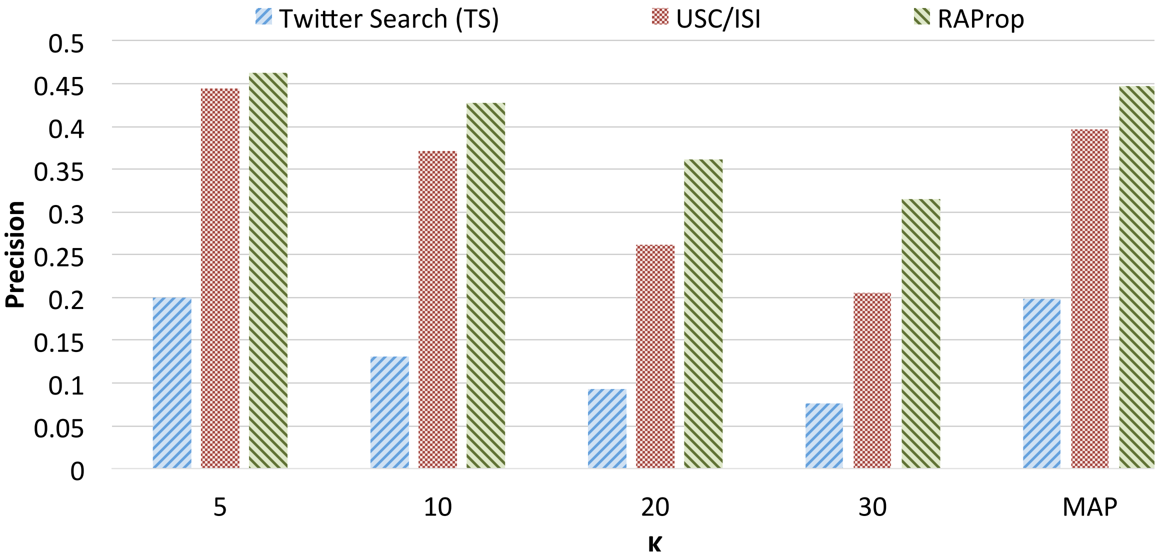


Figure 6.5: *External Evaluation of* RAProp *against Twitter and USC/ISI while assuming a mediator model*

the art ranking method, USC/ISI by achieving higher precision scores for all values of K.

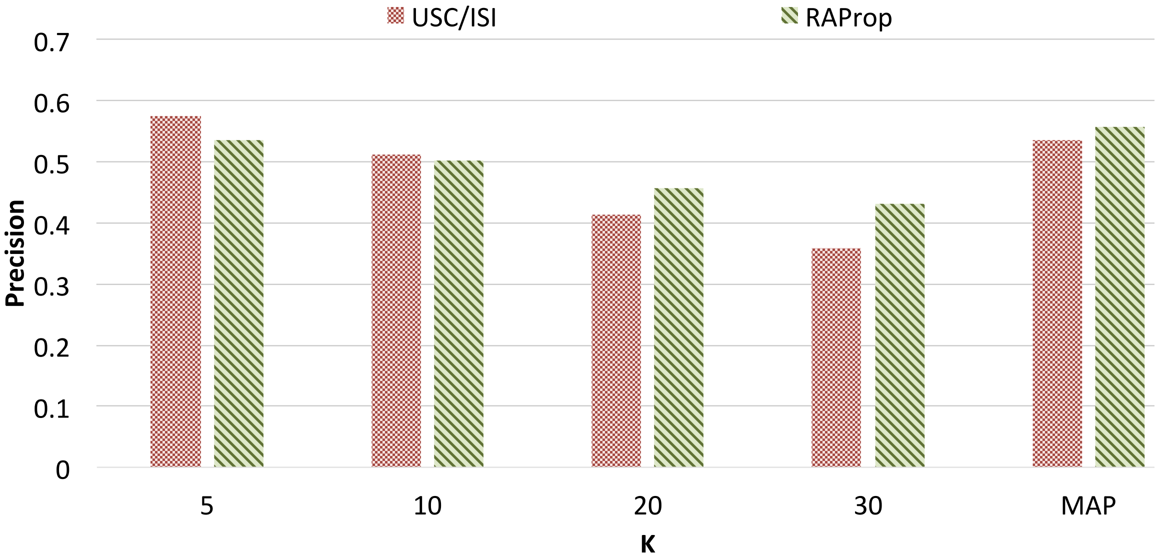


Figure 6.6: *External Evaluation of* RAProp *against USC/ISI on a non-mediator model*

I also compare the precision of my method against the state of the art method, USC/ISI while not assuming to be a mediator. In this method the USC/ISI method is able to index the entire tweet database. The queries are run over this index and the similarity score of each tweet returned by Indri is then combined with other features to finally rank the tweets for that query. I then compare the top-K results of the ranked results with the ranked results of *RAProp*. As shown in Figure 6.6, I noticed that *RAProp* is able to match the precision of USC/ISI at precision at 10 and get higher precision values than the state of the art for higher values of K. Also, I am able to achieve higher MAP values than the USC/ISI ranking. This proves that we are able to rank more relevant results higher in the ranking than USC/ISI ranking.

6.5 Discussion

We have seen that my method *RAProp* achieves higher precision scores that other approaches mentioned as well as the current state of the art. In this section I hypothesize why my method works better than other methods and baselines considered. I believe that ranking tweets should be based on the content of the tweets. But due to the real-time nature of twitter I have no authoritative source that may be used to measure the trustworthiness and relevance of the tweets. Thus I fall back to the belief that popular users produce trustworthy tweets. I use the tweets from popular users as seed nodes that are considered to be trustworthy. Since, I try to assess the popularity of a user in real-time I try a Machine Learning Method that tries to trustworthiness of each tweet. Since I believe the popularity of a user is not a binary assignment, I train the machine learning to assign a score between 0 and 1. The popular nodes although are trustworthy, they are not the only trustworthy nodes in the dataset. There are tweets that from lesser popular users but still contain trustworthy content. I use the inter-tweet agreement as a measure to propagate this trust from the popular tweets to tweets that contain the same or similar content but are from lesser popular users. I then rank the tweets based on this popularity score which I call as *Reputation*

Score.

Let me consider an scenario of a how my method works. Consider the agreement graph, for the query *"Keith Olbermann new job"* from my dataset,shown in Figure 6.7. The nodes in grey circles represents the tweets that are considered relevant, the nodes in white triangles are the ones that are irrelevant by the gold standard. And the nodes in yellow trapezium are the ones that are not present in gold standard and hence I do not know if they are relevant or not. The node size represents the Reputation Score assigned by the Random Forest. The agreement between two tweets is represented by the edge width.

I can see that the relevant tweets form clusters and has high agreement with each other. Likewise the irrelevant tweets form their own clusters. Hence, agreement alone does not help me distinguish between the relevant tweet cluster from the irrelevant tweet cluster. I notice that the relevant clusters have atleast one tweet that has high Reputation Score. Since, other tweets have high agreement with this tweet the propagation of the Reputation Score in the relevant tweets cluster increases the Reputation Score of all the other tweets that agree with that tweet. While in the irrelevant tweet cluster since the Reputation Score of any of the tweet is lower than the relevant cluster, the propagation of that still keeps the Reputation Score of the propagated tweets to be still lower than the relevant tweet clusters. This makes my Ranking rank the relevant tweets higher in its ranking compared to the irrelevant tweets.

In order to analyse the performance of my algorithm over various queries, let me pick a selection of queries where my algorithm does really well (Table 6.1) and where my algorithm does not perform well (Table 6.2). I notice that the queries where I perform really well have a higher percentage of relevant tweets in the data set,*R*. Although the increase in relevant tweets is marginal, I believe that having the higher number of relevant tweets

Query	P@5	P@10	P@20	P@30	% of Relevant Tweets
Haiti Aristide return	1.00	1.00	0.90	0.767	0.14
Pakistan diplomat arrest murder	1.00	0.90	0.85	0.86	0.17
Toyota Recall	1.00	1.00	1.00	1.00	0.34
release of ”The Rite”	1.00	1.00	0.95	0.76	0.12
Emanuel residency court rulings	1.00	1.00	0.90	0.76	0.17
Kucinich olive pit lawsuit	1.00	0.80	0.85	0.73	0.11

Table 6.1: Precision Values of the best performing queries

Query	P@5	P@10	P@20	P@30	% of Relevant Tweets
2022 FIFA soccer	0.00	0.00	0.00	0.00	0.04
Mexico drug war	0.00	0.00	0.10	0.10	0.05
Egyptian evacuation	0.20	0.20	0.10	0.07	0.06
Holland Iran envoy recall	0.00	0.00	0.00	0.00	0.05
State of the Union and jobs	0.00	0.10	0.10	0.07	0.01

Table 6.2: Precision Values of the worst performing queries

makes my method perform better. I believe that the additional number of relevant tweets leads to the existence of more reputed tweets in each cluster and there by helping me rank the relevant tweets higher by the propagation of the score to tweets that agree to it.

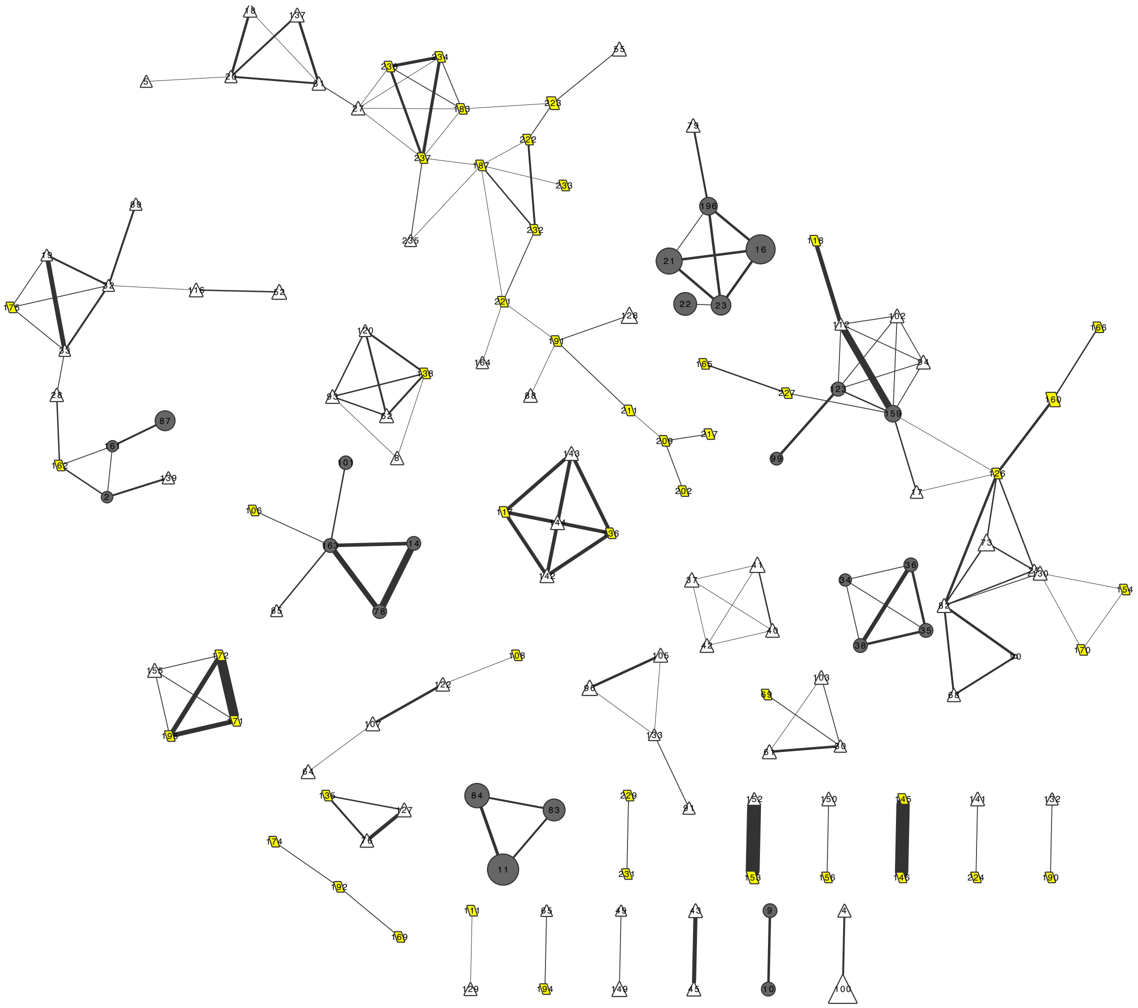


Figure 6.7: *Tweet Agreement Graph where the nodes represents the tweets and edges representing the agreement between tweets. The edge width represents amount of agreement and node size represents the Reputation Score.*

Chapter 7

RELATED WORK

Although ranking tweets has received attention recently (c.f. [24, 21]), much of it is focused only on relevance. Most such approaches need background information on the query term which is usually not available for trending topics. A quality model based on the probability of re-tweeting [9] has been proposed which tries to associate the words in each tweet to the re-tweeting probability. I believe that this approach would not be effective in scenarios where the query brings in more opinions and less authoritative tweets. There are also multiple approaches [22, 13, 19, 18] that try to rank tweets based on specific features of the user who tweeted the tweet. These methods are comparable to the Feature Score (*FS*) method. My approach complements these by adding trustworthiness of the tweets to the ranking algorithm, and can thus be seen as folding many of the features from previous work into a ranking algorithm. Ranking using the mentioned Web Page as a part of the tweet have been considered [20]. I believe that adding web page content to the tweet dilutes the content of the tweet and hence ranking would be based solely on the content of the web page. Hence, the ranking would degrade to ranking web pages.

The user follower-followee relation graph has been used to compute the popularity and trustworthy of a user [8, 29, 1]. These approaches have no predictability when it comes to a user that was not part of the data set on which the popularity was found. They also take much longer for a change in the relation graph to reflect in the popularity score as the algorithm needs to be run over the entire follower-followee relation graph so as to get the new popularity values. Credibility analysis of Twitter stories has been attempted by Castillo et al. [7], who try to classify Twitter story threads as credible or non-credible. My problem is different, since I try to assess the credibility of individual tweets. As the feature space is much smaller for an individual tweet – compared to Twitter story threads – the problem becomes harder.

Propagating trust over explicit links has been found to be effective in web scenarios [6, 17]. I cannot apply these directly to micro-blog scenarios as there are no explicit links between the documents. Finding relevant and trustworthy results based on implicit and explicit network structures has been considered previously [16, 5]. Real time web search considering tweet ranking has also been attempted [2, 12]. I consider the inverse approach of considering the web page "prestige" to improve the ranking. To the best of my knowledge, ranking of tweets considering trust and content popularity has not been attempted. Ranking tweets based on the propagated user authority values have been attempted by Yang [31]. Since the propagation is done over the re-tweet graph, I expect tweets from popular users to be ranked higher. I additionally base my ranking on the content and relevance to the query.

Chapter 8

CONCLUSIONS

Twitter and other microblogs have been increasingly used as a source of real-time news. Their popularity and effectiveness have lead search engines and news platforms to use them as a measure of real-time trends and breaking news. The popularity and uncurated nature of these microblogs makes it suspectable to spams. Twitter Search may be considered as a real-time information search on a topic. Hence there is a need to provide results that are highly relevant to the query and that are trustworthy. Twitter defines the recency of the tweet as the measure of relevancy. Hence, Twitter sorts the tweets that contain the query keyword in the level of recency and it counters spam by blocking accounts that are suspected to be malicious in nature. I believe although recency of the tweet may be considered as a factor for relevancy, it shouldn't be the only measure of relevancy. Also, I believe that we need a more granular definition for trustworthy as all non-malicious tweets may not be trustworthy tweets.

I propose a method to rank the microblogs considering the relevance ant trustworthiness of the content. I model the Twitter ecosystem as a three layer graph consisting of user, web and tweet layers. I compute the reputation scores of entities in all the three layers. These scores are transmitted to the tweet layer through inter-layer links, and combined to formulate a single reputation score using a random-forest learner. Finally, I propagate the reputation scores between the tweets through agreement links, thereby leveraging collective intelligence of tweets to compute trust.

My detailed experiments on a large TREC microblog dataset shows that agreement or Reputation Score alone is not able to achieve the same performance as combining them together. I also evaluate the performance of various design choices I made and show that my method performs better than other variants possible. My external evaluations show that the proposed method improves the precision of ranking not only over the Twitter's own ranking but also over the current state of the art ranking algorithm. Although the algorithm was proposed for a mediator model, I show that the algorithm performs better than other baselines even when this assumption is relaxed. I explain why I believe multiple values of propagation of the Reputation Score over the agreement graph may not lead to better results. This intuition was later proved right using the empirical evaluations. I also try to explain why I believe *RAProp* is performing better than other baselines and design choices considered. Thus, I propose a novel ranking approaches for microblogs that considers relevance and trust that achieves better results than current state of the art method on the same dataset.

REFERENCES

[1] M.-A. Abbasi and H. Liu. Measuring user credibility in social media. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 441–448. Springer, 2013.

[2] F. Abel, Q. Gao, G. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. *User Modeling, Adaption and Personalization*, pages 1–12, 2011.

[3] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

[4] R. Baeza-Yates, C. Castillo, V. López, and C. Telefónica. Pagerank increase under different collusion topologies. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pages 17–24, 2005.

[5] R. Balakrishnan and S. Kambhampati. Sourcerank: Relevance and trust assessment for deep web sources based on inter-source agreement. In *Proceedings of WWW*, 2011.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, pages 107–117, 1998.

[7] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of WWW*, 2011.

[8] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *4th international aaai conference on weblogs and social media (icwsm)*, volume 14, page 8, 2010.

[9] J. Choi, B. Croft, and J. K. Kim. Quality models for microblog retrieval. In *Proceedings of CIKM*, 2012.

[10] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IIWeb*, pages 73–78, 2003.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[12] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of WWW Workshop*, pages 331–340, 2010.

[13] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.

[14] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, DTIC Document, 2010.

[15] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. *Cooperative Information Agents VII*, pages 238–249, 2003.

[16] M. Gupta and J. Han. Heterogeneous network-based trust analysis: a survey. *ACM SIGKDD Explorations*, pages 54–71, 2011.

[17] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587. VLDB Endowment, 2004.

[18] L. Jabeur, L. Tamine, and M. Boughanem. Featured tweet search: Modeling time and social influence for microblog retrieval. In *IEEE/WIC/ACM International Conference on Web Intelligence*, 2012.

[19] J. Jiang, L. Hidayah, T. Elsayed, and H. Ramadan. Best of kaust at trec-2011: Building effective search in twitter. In *Proceedings of the 20th Text REtrieval Conference (TREC 2011)*, 2012.

[20] R. McCreadie and C. Macdonald. Relevance in microblogs: Enhancing tweet retrieval using hyperlinked documents. 2012.

[21] D. Metzler and C. Cai. Usc/isi at trec 2011: Microblog track. In *Proceedings of the Text REtrieval Conference (TREC 2011)*, 2011.

[22] R. Nagmoti, A. Teredesai, and M. De Cock. Ranking approaches for microblog search. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010*

IEEE/WIC/ACM International Conference on, volume 1, pages 153 –157, 31 2010-sept. 3 2010.

[23] Twitter death hoaxes, alive and sadly, well. http://nyti.ms/10qVW9j.

[24] Trec 2011 microblog track. http://trec.nist.gov/data/tweets/.

[25] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. *The Semantic Web-ISWC 2003*, pages 351–368, 2003.

[26] Zombie followers and fake re-tweets. http://www.economist.com/node/21550333.

[27] State of twitter spam. http://bit.ly/d5PLDO.

[28] About top search results. http://bit.ly/IYssaa.

[29] Y. Yamaguchi, T. Takahashi, T. Amagasa, and H. Kitagawa. Turank: Twitter user ranking based on user-tweet graph analysis. In *Web Information Systems Engineering–WISE 2010*, pages 240–253. Springer, 2010.

[30] Yammer. http://www.yammer.com.

[31] M. Yang, J. Lee, S. Lee, and H. Rim. Finding interesting posts in twitter based on retweet graph analysis. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1073–1074. ACM, 2012.