Measuring influence by including Latent Semantic Analysis in Twitter conversations

by

Xiaobo Deng

This Master Thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education.

Master Thesis in
Information and Communication Technology

Faculty of Engineering and Science
University of Agder

Grimstad, May 25, 2011

Abstract

As the mount of information is growing in social media, online influence estimation is becoming significant and an time consuming element in social media analytic. In the last few years therefore there have been several algorithmic approaches to automate the estimations. Examples of such algorithms are ExpertiseRank and Klout Score. In this thesis, we propose an online influence estimation algorithm. We name it XRank. XRank is a novel approach to include content analysis into the traditional influence estimation domain. In traditional estimation techniques they mainly use metadata like followers or friends. In our proposed solution, Latent Semantic Analysis(LSA) enables XRank algorithm to have capability of estimating online influence based on given topic. By designing and implementing an algorithm prototype and testing with different dataset sizes, vocabulary size and vocabularies with different topics, we measure how these parameters affect XRank result. We also compare the XRank estimation result with another online influence estimation algorithm called Klout Score.The testing results suggests that XRank shows satisfactory performance based on given topic. We believe that the result will provide a new point of view to online influence estimation.

Preface

This master thesis is submitted in partial fulfilment of the requirements for the degree Master of Science in Information and Communication Technology at the University of Agder, Faculty of Engineering and Science. The project is supported by Integrasco A/S, who has provided data material and insight for the various simulations performed in this study. This work was carried out under the supervision of Tarjei Romtveit at Integrasco A/S, and co-supervisor associate professor Ole-Christoffer Granmo at the University of Agder, Norway.

First I would like to thank my supervisor Tarjei Romtveit, for his assistance and guidance throughout the project period. He provided a lot of valuable advices and, for many times, rescued me from wrong directions so that I can focus on my target. And he also gave me plenty of suggestions about writing thesis report. Without his help I probably never accomplished my master thesis. I also want to thank professor Ole-Christoffer Granmo for his assistance and support. He helped me a lot regard to the algorithm design and provided some impressive ideas. I also want to thank my colleges in Integrasco who offered help to me when I was looking for support. A special thank to Jaran Nilsen who approved me to halt my work in Integrasco for a period of time so I can focus on writing my master thesis.

Grimstad, May, 2011
Xiaobo Deng

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online estimation is becoming a epidemic research with the emerging of social media in recent years. Researchers have proposed several estimation algorithms to estimate online influence. Algorithms like ExpertiseRank are utilizing relationships between users as its estimation parameter. Online influence derived by ExpertiseRank is determined by other users who are connected. Other algorithms such as Klout Score utilizes dozens of variables to evaluate online influence. These variables include followers and unique commenters ect. However, neither of them considers the user generate content in social media.

In this thesis, we proposed XRank algorithm to estimate online influence in Twitter. By introducing LSA technolgy, XRank has the capability of knowing how close one document related to a given topic. Then we combine retweet* to estimate online influence.

1.1 Background and motivation

As the advent of internet, social media enables people to create and share content. Amount of such content are posted and discussed through social media networks[30]. Since social media encourage contribution and feedbacks from anyone who interested, types of content range from art and business to technology and entertainment are generated by users everyday. Social media services also enable communities to have conversation with rare barriers[18]. Known as a common characteristic of social media, connections(relationships) between people are patterns of forming these people as a society. Throughout direct or indirect connections, content propagate to other people all around the world

* retweet is an action of sharing a tweet in Twitter

quickly. Moreover, social media enables companies to listen to their customers' feedback about their brand. Companies need to know what people are saying to their product and how such opinions could impact their business. Such commercial demands induce the appearance of social media analytic[26]. Social media analytic is concentrating not only on what is being said about products, but also on who is saying. A small portion of social media users have capabilities of persuading others and creating leading trends. We call them *influential* in social media. Research states that 80% of consumers trust advice form friends online and one in three internet users looking for help from online communities to make purchase decisions[39, 11, 19]. To find influencers are valuable for companies and organizations to improve their products and services and even to design the best advertisement strategies[8].

Focus on discovering influencers in social media has been a popular subject for researchers the last few years. The researchers have proposed several algorithms to measure online influence. In 2007, Jun Zhagn et. al.[41] suggested ExpertiseRank to estimate online influence on question/answer forums. ExpertiseRank is an PageRank-like algorithms and utilizes user's relationships to rank online influence. Another commercial ranking algorithm is Klout Score[*]. Klout Score algorithm provides an overall online influence for users on Twitter[†] and Facebook[‡]. Klout Score algorithm utilizes many variables in social media. It includes both relationships between users and other parameters such as activities as its parameters.

Different from traditional estimation algorithms, we intend to propose an algorithm that involves content analysis in its influence estimation procedure. The reason is that we believe that user generated contents are related to users influence. The fundamental idea user generated text content represent what the user is saying. In addition, we also interested in what is the content writes about. That means we also concern what topic is the user talking about. Based on this knowledge, our algorithm will be capable of ranking users in different topics. Document classification methodology furnishes a technique called LSA to achieve this target. By constructing a term-document matrix according to set of text content(document) and applying Singular Value Decomposition(SVD) on this matrix, LSA maps real document into a multi-dimensional document space. Moreover, given vocabulary contains the keyword related to specific topic would be mapped to document space as well. Next the similarity measurement is employed to find similarities between documents and topic. Influence on social media is not only related to content, but also regards to the relationships and interactions between users. So we intend to introduce other variable to

[*] www.klout.com/kscore [†] www.twitter.com [‡] www.facebook.com

XRank.

Among several social media platforms, Twitter is a very popular social network applications. Twitter is a real-time information application that enables users to communicate by sending and receiving short messages called *tweet*. Each tweet is in 140 characters limit. People can share their ideas by posting tweets or retrieving and following prevalent subjects. Nowadays, Twitter changes the way we are communicating. On twitter, information is flowing faster than that on traditional media and it simplifies the conversations between users[10]. Another important characteristic worth to mention is that twitter is driven by influencer. These influencers generate plenty of valuable ideas and these ideas are wildly spread very widely and sometimes very fast.

Since it is difficult for people to be aware of the influence of users on Twitter in terms of number, people try to seek out ranking algorithms that represent online influence intuitively, with a mount of valuable variables. Meeyoung et. al.[8] speculated that indegree, retweet and mention could be very interrelated to online influence in Twitter[*]. By utilizing about 1.7 billion tweets from Twitter, Meeyoung et. al found that retweet influence has tight correlation with mention influence. Moreover indegree influence was not related to other measures. Another online influence evaluation algorithm for Twitter user called Klout Score is utilizing more variables, and it will be discussed in 2.2.2.

In this thesis, we proposed a new algorithm to estimate online influence in Twitter. We name it XRank. XRank algorithm utilizes retweet as one of its variables. Moreover, XRank includes content analysis by applying LSA technology.

1.2 Thesis definition

We formulate the thesis definition in the following fashion:

In this thesis, we want to measure online influence on specific users in social media by utilizing techniques in Latent Semantic Analysis. In order to archive this target, we would like to analyze the content of conversations posted by users in social media and deduce an influence of the different participating users numerically. We intend to design and implement a prototype estimator that will perform several estimations against real social media discussions and create a derivation process to obtain users' online influence.

[*] Indegree, also seen as the number of followers, reflects the popularity of user, retweet count represents the value of the tweet content and mentions indicates the worth of the user name.

1.3 Research questions

1.3.1 In what degree can we use LSA technology to estimate online influence?

This is the main question of this master thesis. On social media communities, users contribute amount of contents. Contents and user's profiles are emerging together to become a virtual society. In this virtual world, users have their own influence as the real world. The initial idea of this master thesis is to research how user generate content affect online influence. Based on our investigation, LSA can be use to analyse text content which are common found on social media. We intend to research whether the content of text would have tight correlation with online influence. It means that if we include LSA to help to evaluate online influence, we want to research the performance and accuracy of XRank.

In what degree of changing the size of vocabulary affect the XRank result?

When applying LSA to test document, LSA will create document space from the tested documents. In this document space, all documents are represented as an multi-dimensional vector. Then a fake document is created from a vocabulary which contains the query content belongs to a specific topic. This fake document also would be mapping to the document space in forms of a vector with the same pattern as test document has. LSA then can compare the similarities between test documents vector and fake document vector. A substantial aspect here is how large should this vocabulary be and how the vocabulary size would affect the rank result derived from the XRank algorithm.

In what degree can XRank distinguish users influence based on topic?

The task of XRank is estimate online influence on social media. Which means, XRank should be able to find leading influencers from different topics such as users can range from technology and business to health and design. Since LSA can reveal the latent semantic meaning of words and meaning of documents. We can create our word or vocabulary, and to see which document(s) have close meaning to the word(s) we created. For instance we create a vocabulary for art. When we are applying LSA to set of documents, we would like to see if XRank can find user talking about art.

In what degree of changing the size of dataset affect the XRank result?

The principle of LSA is analysing content to get relationships between terms and document or between document and document. Therefore a very critical aspect of applying LSA is how large the documents would be. Since the meaning of the paragragh is determined by average meaning of the word and the word meaning is determined by average meaning of the document[24]. The size of the document would influence the meaning of the document. Therefore, document size would affect the result of content analysis. We wonder how XRank is affected by document size.

In what degree do XRank correlate with Klout Score?

Klout Score is a business product of evaluating online influence on Twitter [*] and Facebook [†]. By utilizing over 35 variables to get an overall online influence. In this thesis, we will apply XRank algorithm to Twitter data in two different topics and obtain two estimation results. We intend to compare these two results with Klout Score to disclosure the characteristic of XRank and Klout Score.

1.4 Claims

In this thesis we claim that XRank algorithm demonstrates its advantage against transitional online influence estimation methods . We attribute this superiority to the capability of content analysis provided by LSA. We also claim that XRank algorithm can be used to estimate online influence more than in Twitter. XRank can be used in other social media communities such as Facebook by simply adapting few parameters.

1.5 Contributions

In this thesis, we proposed a new solution to rank online influence on social media by including LSA. Basically, we would like to measure the impact value of user generated content as a new estimation variable. In this project, LSA is not only used to obtain the correlations between user generated content and given subject/topic, but also utilized to estimate how much are talk about given subject/topic. Based on both correlations and

[*] www.twitter.com [†] www.facebook.com

weight of user generated content regard to given topic, LSA derives content impact value. Then combing with another social media metadata, XRank derives a online influence result for users.

1.6 Target audience

The target audience of this thesis is anyone who interested in influential ranking or influence evaluation on social media. This thesis also involves knowledge about document classification hence it is ready for people who concerns to document classification as well. Because this thesis proposes a solution based on LSA, people who interested in LSA may also read this thesis. Since LSA is related to matrix operation, the reader should have fundamental knowledge about linear algebra and matrix manipulation.

1.7 Report outline

The rest of this report is organized as following description: Chapter 2 introduces the basic concept of social media, and influential definition on social media, as well as the influence measurement. This chapter also retrospect two types of traditional measurement methods, ExpertiseRank algorithm and Klout Score. Chapter 3 demonstrates the history of natural language processing and document classification. Then we depict several traditional methods which are used to measure document similarity. Chapter 4 describes the concept of LSA and shows how LSA be used to document classification. Chapter 5 proposed a solution named XRank algorithm which provide a new approach to measure online influence. In this chapter, we explain XRank algorithm in details. Chapter 6 describes the experiment setting, elaborates several tests cases and demonstrates testing results. Based on test results, we make some discussions to answer the research questions in Chapter 1. Chapter 7 makes a conclusion and suggests several aspects can be further work.

Chapter 2

Social Media and Online Influence Evaluation

This chapter briefly introduces social media conception, as well as online influencer definition in social media network. This chapter also exhibits two traditional approaches of evaluating online influence in social media.

2.1 Social media and online influence evaluation

2.1.1 Social media

Along with the prevailing of modern internet development, plenty of web services are invented. Some of web services provide online communities which are enabling people to communicate on internet. These virtual communities are generally called social media. As Antony MayField [18] saying, social media is actually about being human beings, to fulfil the need of sharing ideas, cooperating and collaborating, thinking and debating and finding people who can be friends. Usually by creating a profile to join a network, people are able to express their opinions or communicate with each other in the same network. On social media, people are encouraged to express their ideas.

Nowadays, there are several types of modish social media includes blogs, forums, social networks, podcast and microblogging. Example of famous social network is Facebook. Facebook is the largest friend network on internet. On Facebook, you can for example post your private or public photos and blogs and update you latest relation status. Additionally, you can connect to your friends in real life or get to know new friends. After logging to you page, you can easily know what your friends really doing, and comment their photos and status. These interactions are easily performed and appeals to a broad audience of the population.

2.1.2 Influence definition in social media

As we discussed in section 1.1, there exists influencer on social media who leads the trends and have prestiges. The studying of influencer circumscription will help us to understand the reason of trend prevails or innovation are adopted faster and help advertisers to design impact campaigns[8], hence help us to design a influence estimation algorithm. Unfortunately, there is no unitive opinion about influencer on social media. Roger(1962) believes that influencers are people who can lead the trends, more innovative and they are always in the center of network[29]. De-emphasis the role of influencer, anther opinion of factors of determine influencer are interpersonal relationship in ordinary users and readiness of a society to adopt an innovation. Empirically, influencers nowadays are more like people have high level expertise than the rest, and they are gladly to help others by answering questions and providing suggestions. These influencers always have good reputations on the communities and people would like to listen to their ideas.

2.2 Traditional evaluations

To estimate online influence in social media, several algorithms are proposed. Among them two algorithms are worth to mention. One is ExpertiseRank and another one is Klout Score. The former algorithm is Page-Rank like algorithm and Klout Score represents the new thought of evaluating online influence.

2.2.1 ExpertiseRank

ExpertiseRank [41] was first proposed by J. Zhang el at. in 2007. ExpertiseRank is a PageRank-like algorithm which tested on questions/answering forums such as Yahoo!Answers [*]. Online communities have a thread structure. One thread is stared with a topic or a question, people who interested in this topic will join this thread by replying previous post in the

[*] http://answers.yahoo.com/

same thread. The posts in one thread form a typical conversation in social media. ExpertiseRank algorithm uses this thread structure as its foundation. ExpertiseRank algorithm assumes that the user who answer the question has a higher expertise level than the one who post the question. The more the user helping, the higher expertise rank the user has. In contradiction, the more people helped the lower expertise level the user has. Additionally, the expertise level propagate through the expertise rank network. If user A can answer the B's question and B can answer C's question, then A has a higher expertise level than C has.

The ExpertiseRank algorithm is introduce in[41] as:

> Assume User A has answered question for users $U_1 ... U_n$, them the EpxertiseRank(ER) of user A is given as follows.
>
> $$ER(A) = (1-d) + d(\frac{ER(U_1)}{C(U_1)} + ... + \frac{ER(U_n)}{C(U_n)}) \tag{2.1}$$
>
> C(Ui) is defined a the total number of users helping U_1, and the parameter d is a damping factor which can be set between 0 and 1. we set d to 0.85^2 here. The damping factor allows the random walker to 'escape' cycles by jumping to a random point in the network rather than following links a fraction(1-d) of the time.

ExpertiseRank is utilizing the user relationship to propagate influence through out the network. Guha et al.[16] research the trust propagation problem and distrust among Epinions users.[20] Actually, this kind of user relationship is also used to find high-quality content in social media.[1]

2.2.2 Klout Score

About online influence, Klout score team believes that influence is the ability of driving people to reply, to retweet, to comment and to click[21]. Klout socre utilizes over 35 parameters mainly to evaluate Real Reach, Application probability and Network Influence. True Reach is the evaluation factor about the size of engaged active audience. Application probability is checking the frequency of content interaction. And Network Influence the parameter of the level of engaged audience. The final Klout score rang is between 1 and 100 to represent online influence in Facebook and Twitter. In order to get True Reach Value, followers, mutual follows, friends, total retweets, unique commenters, unique likes follower/follow ratio, followed back, mention count list count and list followers count are collected to become the parameters. Because Application probability utilizes indexes such a unique retweeters and unique message retweeted to estimate engagement of one user. As network influence, Klout score focus on the influence of engaged audience.

Chapter 3

Document Classification

3.1 Natural Language Processing

In 1950s, after the first computer in the world was invented, scientists were exited to develop the potential capacity of computer. Scientists developed various applications such as machine translations, artificial intelligence, speech recognition and text document classification/categorization. Research about Natural Language Processing (NLP) starts with ”Computing Machinery and Intelligence”[36], a paper published by Alan Turing in 1950. The paper set a criteria for artificial intelligence. This criteria, called *Turing Test* (TT), prescribed a method to access whether or not a machine can think like human[33]. Originally, in Turing’s paper, TT is a Imitation Game(IG). In his paper, he also represented TT in another manner: *”Can machines communicate in natural language in a manner indistinguishable from that of a human being?”* [33]. NLP is trying to enable computer to understand and manipulate spoken and written human language. On this purpose, different types of knowledge are involved like information science, linguistics, mathematics and electronic engineering.

In 1980s, people introduced machine learning to NLP systems. Before machine learning was utilized by NLP algorithm, processing rules were configured by human. Machine learning enables NLP systems to learn from human-labelled data and unlabelled data. Machine learning can be divided into several types. Three typical are supervised learning, unsupervised learning and semi-supervised learning. NLP systems with supervised learning will be trained before they are employed. The attributes training data are annotated by human. In some cases, the accuracy and performance of the NLP system is proportional increased with the size of the training data. However, in some case labelled instances are difficult, or may be time consumption[42]. In some special cases, it it even impossible to label data attribute. For example, human are unable to explain and label the expertise level of audio sample in speech recognition. Different from supervised learning, unsupervised learning systems don't need the labelled training data. It learns from the data to find most common rules for all data. Semi-supervised learning systems learn from both label and unlabelled data, to build a better rules and better classifier.

Nowadays, NLP systems are wildly used in information retrieval(IR) area and becomes a much more complicated and advanced method. Types of retrieval algorithms ranging from simple to complicated such as Boolean expressions, Vector Space Model(VSM), TF-IDF and Naive Bayes Classifier are proposed based on NLP principles. Boolean expression are keywords combined with AND, OR or NOT. Then the expression is to match with document collections according to the logic generated by logic conjunction. VSM, TF-IDF and Naive Bayes Classifier algorithm will be discussed in the following sections.

3.2 Document classification

Information on internet can be in different forms such as audio, video and document. Documents take a large proportion among all internet resources. Document classification is a sub-theme of NLP, aiming to partition unstructured document into groups with similar properties. From perspective of prior conditions, there are two variants – document clustering and document categorization (document spotting)[17]. In document clustering problem, properties are not known advanced. Documents in collection are organized into groups that documents are similar to each other and dissimilar to those in other groups[2].Because of this reason, document clustering is a unsupervised learning. On contrast, in document categorization problems, the properties (the classes) are known advanced and all document are assigned to these classes.[2] As an example, email spam detection is a document categorization application with two known classes – regular email and spam email. In next we will introduce three typical document classification methodologies that represent different approaches of NLP.

3.2.1 Vector Space Model

Vector space models (VSM) was first represents by Slaton et. al. in 1975[32]. VSM is an algorithm that represents document as vector of identifier. Each of documents is represented by a vector. The vector consists of a list of weight of unique term in the document. Each dimension demonstrates the term in document. If the term exists in the document, the weight will be a non-zero value. The term can be different forms such as a word, a phase or a sentence according to applications. The weight in the vector is the number of times of term occurrence in document. A docuument can be represented in 3.1:

$$D_i = (d_{i1}, d_{i2}, d_{i3}, ...d_{in}) \tag{3.1}$$

d_{ij} is term weight. While queries (fake document) are also similarly represent in the same way. Given constructed document vectors and query vectors, several similarity measurement methods such as Cosine similarity or Euclidean Distance can be applied to get coefficient similarity. Coefficient of similarity proclaims the degree of similarity of two documents.

VSM provides a simple and easy way to evaluate document similarities. However, it has some weaknesses. VSM counts number of terms that is not relevant to the meaning of document such as ”the” and ”a”. At the same time, long document contains more items than short document. Thus long document obtain higher coefficient similarity value, not because of the content but the length of document. Since search term has to be exactly match document term, the substring of word will cause an error called ”false positive match”. Based on the same reason, documents with same context but with different vocabulary will not associated, resulting a ”false negative match”.

3.2.2 TF-IDF

Douglas W. Oard noticed that documents are described by higher frequency term, also by the low frequency terms [38]. VSM just simply count the term frequency(TF). Hence low frequency terms have disadvantages. Term Frequency-Inverse Document Frequency(TF-IDF) utilizes new factor which is called inverse document frequency(IDF) to assign more weight to rare terms and decrease weight to meaningless terms. Since long documents contain more words, simply counting term occurrence times is unfair to short documents. To eliminate this weakness, term frequency is introduced in 3.2.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.2}$$

$n_{i,j}$ is the term counts in document, $n_{i,j}$ divides sum of all occurrence number to get term frequency. While inverse document frequency measures importance of terms occurs in document. It is represented in equation 3.3.

$$idf_i = \log(\frac{|D|}{|j : t_i \in d_j|}) \tag{3.3}$$

$|D|$ is the total number of document in corpus. $|j : t_i \in d_j|$ is the number of all document where terms exists. To avoid division-by-zero caused by no term appears, usually plus 1 with $|j : t_i \in d_j|$. Finally the TF-IDF weight is obtained by $tf_{i,j} \times idf_i$. TF-IDF weight is often used by VSM to improve the IR performance.

3.2.3 Naive Bayes Classifier

Naive Bayes classifier is a type of supervised classification method which applying Bayes' theorem. The underneath of Naive Bayes classifier is a condition model shown in equation 3.4.

$$P(A|B) = \frac{P(B|A)P(A)}{PB} \tag{3.4}$$

Here, $P(h)$ is the prior probability of hypothesis h, $P(o)$ is the prior probability of observation, $P(h|o)$ is probability of h given o and $P(o|h)$ is probability of o given h. When applying the condition model to application, there is an important assumption: hypotheses are exclusive and exhaustive. That means all conditions are considered and listed. Moreover, no overlap among all listed conditions. Based on label data, correct training strategy will lead to a precise classification result. The document classification applications try to get probabilities for all presetting hypothesizes. And the most probable hypothesis is the final result. However, there still exists some misclassification either because of the training or the threshold set in the application.

Naive Bayes classifier algorithm is supervised document classification method. The advantage of such methods is that the result can be improved by continuous learning. However, the disadvantages are obvious as well. Supervised classification is usually designed and improved for specific application. People have to design different strategies to apply different requests and domains. Another weakness is that training data have to be label by external mechanism. Sometimes it is easy to set label for objects, but sometimes it is difficult or impossible to set. For example, general application like set the professional level of a passage, it is not so easy to label levels. At this time, we have to take other approaches such as unsupervised classifications which don't need any external interference. In IR field, LSA is a typical unsupervised classification and we will introduce LSA later.

3.3 Normalization

3.3.1 Stop word

Document often contains some meaningless words frequently. It is commonly believe that these words don't contribute to the meaning of the documents[25]. The reason why they exist is partly because of grammar needs[43]. Example of stop word can be like ”a” ”and” ”the” ”on”, et. When applying IR algorithms to documents, these words better to be removed from the document, both for eliminating unexpected error and saving computing resources.

3.4 Similarity measurement

In IR fileds, several document classification algorithms have to represent documents as a form that can be estimate by computer. The most popular routine is mapping documents to vectors. Vector Space Model , TF-IDF and LSA are algorithms that utiliz such approach. By utilizing document vector, it becomes possible for computer to compare document with different measurement methods. Measurement methods to measures two vectors includes pivoted normalization, simple vector product, cosine similarity Euclidean distance measurement. Each of them has adequate to specification situations where they come to their advantages. In this thesis, we use cosine similarity as our similarity measurement technique.

3.4.1 Cosine similarity

Cosine similarity is a method that calculates the cosine value between two vectors. Cosine value have to be ranging from -1 to 1. Where -1 indicates the angle is 180 degree and 1 means zero degree for two vectors. In another word, value 1 suggests that two vector points to the same direction and vice versa. Cosine similarity utilizes following formula to get the similarity value:

$$similarity = \cos(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(Ai)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

As mentioned before, both VSM and TF-IDF can abstract documents as document vector. Cosine similarity between documents shows the angle of two document vector in document space. The limitation of cosine similarity is that, it only reflects the angle of two vectors but not the length of each vector. LSA will be helpful to resolve this problem and will be demonstrated in the next chapter.

Chapter 4

Latent Semantic Analysis

This chapter discusses deficiency of traditional term-based algorithms. We then introduces LSA algorithm. We also reveal underlying mathematical machinery of LSA.

4.1 Latent Semantic Analysis

LSA is an unsupervised analysis technique of representing similarity of expected contextual usage of words in passages of discourse[24]. In 1988, LSA was patented by Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum and Lynn Streeter. LSA is also called latent semantic indexing(LSI) in its implementation in IR. LSA represents the meaning of a word as the average meaning of the passage(s) and the meaning of (a) passage(s) as the average meaning of all words contained in (a) passage(s)[24]. This signifies that LSA reveals word-word relations which are similar to human recognition. Similarity measurement of LSA is not just the statistics of word occurrence count, but also inferred substantial relationships and meanings. Empirically, when human read or write passage or document, the choose of vocabulary reflects the meaning of the document, words may have specific meanings in individual document or passages. LSA is capable of extracting such latent meanings from passages.

Traditional term-based IR algorithms match exactly the same word as that in query among all documents. Because of the polysemy and synonymy, types of traditional algorithms can not return the results(documents) that are without the words in query. For example, if user wants to retrieve documents related to *mobilephone* by utilizing query ”laptop”, and only documents contains word ”mobilephone” will be matched. However, document contains ”smartphone” also could be the result the user wants. In Deerwster’ paper [12], a retrieval example shows the weakness of traditional retrieval algorithms. As shown in Figure 4.1, each row represents a document and x represents the word occurrence in each document. With query string of *”IDF in computer-based information look-up”* , x with start indicates that words in document also appears in query. ”R” in REL column indicate Doc 1 and 3 are relevant to the user’s query while ”M” in MATCH column suggest that Doc 2 and 3 are the retrieval result by traditional algorithm. Instead of return Doc 1 and 3 which are user actually want related to ”computer and information”, traditional methods return Doc 2 and 3. Doc 1 is mismatched and Doc 2 is ”negative” matched.

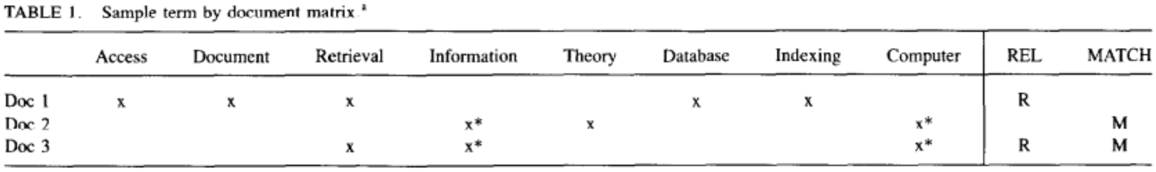


TABLE 1. Sample term by document matrix ^{a}

	Access	Document	Retrieval	Information	Theory	Database	Indexing	Computer	REL	MATCH
Doc 1	x	x	x			x	x		R	
Doc 2				x*	x			x*		M
Doc 3			x	x*				x*	R	M

Figure 4.1: Sample of traditional retrieving algorithm procedure and result

LSA can be viewed as two ways:(1) LSA can obtain the approximate estimation of contextual usage of words in text and (2) be a model of processing and representing substantial meaning of passages[24]. For the first view, LSA extracts word-word, word-passage and passage-passage correlations into a semantic space. Words with similar usages way probably have similar literally meanings[24]. As the model of externalizing underlying meaning of passages, LSA compares the similarity of document vector in document space. These two views are actually inter-osculated to each other. The words meaning not only determined by the semantic of themselves, but also depend on occurrences of other words in passages. In another word, the meaning of passage also refers to the usage of all word it contains.

Schreiner et. al. has shown that LSA can be used to assess student knowledge[35]. Their research indicates how LSA grades students essays and how LSA classifies appropriate instructional text, by comparing the cosine similarity between vector abstracted from an essay written by student with one or more document vector abstracted instructional text. Based on Schreiner’s research, Rehde et. al. continued the study of finding answers to several research questions include that does LSA depends merely on technique words(vocabulary) [40]. In another words, if a student creates a bag of technique words instead of writing an essay, would LSA does equally well as before. Their result suggests that creating a bag of technique words might effective although it difficult for one to create a technique word list extracted from text copra[35]. In this thesis, a similar question would be researched and answered, which is how does the query(technique word) effects the performance of XRank which includes LSA.

Implementing LSA contains several steps include building term-document matrix, applying Singular Value Decomposition(SVD), similarity measuring, etc. Details of these procedures will be discussed in the following sections. A simple mathematic example is demonstrated in Appendix B.

4.2 Construct Term-Document matrix

LSA analyzes the words in corpus to construct a term-document matrix to denote the relationship between term occurrences in documents. In term-document matrix, rows of matrix represent terms and columns stand for different documents. Element of matrix shows the number of times the term occurrence in each document. The elements are subjected to a preliminary transformation, in which each cell frequency is weighted by a function that expresses both word's importance in the particular passage[24]. The term-document matrix not only shows the term frequency in each passage, but also represents the frequency of between different terms. A typical term-document matrix is represented in Figure 4.2.

$$\begin{array}{cc} & \begin{matrix} \phantom{x_{1,1}} & \mathbf{d}_j & \phantom{x_{1,n}} \\ & \downarrow & \end{matrix} \\ \mathbf{t}_i^T \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \end{array}$$

Figure 4.2: Term-document matrix in LSA

A row in matrix $t_i^T[x_{1,1}\,, x_{1,2}\, x_{1,n}]$ denotes the relation between documents from d_1 to d_n. A column in matrix dj demonstrates the term relationship in document d_j. As discussed in chapter **??**, based on the consideration of words appear in many documents should take lower weight and words appears in rare document should be set higher, the cell value of matrix can be replace by TF-IDF value.

4.3 Apply Singular Value Decomposition

LSA applies SVD to term-document matrix which was build in previous step. The procedure of decomposition original term-document matrix into three matrixes: U, S, V^T called *singular value decomposition*.

$$A = U\Sigma V^T$$

Assume that M is an $m \times n$ matrix, then U is an $m \times m$ unitary matrix represents the word usage meaning of text corpus. Σ is an $m \times n$ diagonal matrix consists of non-negative real numbers. V is an $n \times n$ matrix which stands for the document matrix which represents meaning of text corpus. Matrix V^T represents the documents matrix in which each row represents one document vector on behalf of one document. The columns of U and V are called left singular vectors and right singular vector of A, respectively. SVD is unique depends on the original matrix and sign permutation[12].

$$A = U\Sigma V^T = \left(\begin{array}{c|c|c|c} & & & \\ U_1 & U_2 & \cdots & U_m \\ & & & \end{array}\right)\begin{pmatrix}\sigma_1 & 0 & \cdots & 0\\ 0 & \sigma_2 & \cdots & 0\\ \vdots & & \ddots & 0\\ 0 & \cdots & & \sigma_n\\ 0 & \cdots & \cdots & 0\\ \vdots & & & \vdots\\ 0 & \cdots & \cdots & 0\end{pmatrix}\left(\begin{array}{c}(V_1)^T\\ \hline (V_2)^T \\ \hline \vdots \\ \hline (V_n)^T\end{array}\right).$$

Figure 4.3: Singular Value Decomposition matrix

By convention, non-zero values in matrix Σ are sorted by decreasing order:

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq \sigma_n \geq 0$$

4.4 Rank approximation

Piratically, a low rank approximation matrix will replace the original matrix based on SVD of A[14]. Rank approximation is to remove the extraneous information from original dataset[4]. Rank approximation is applied by different types of applications such as data statistics [15, 5, 3], data processing[28] and seismic tomography[6, 34].

$$A = U\Sigma V^T \approx U_k \Sigma_k V_k^T =: A_k$$

Since values in matrix is decreased ordered which is discussed in 4.3, where Σ_k consisted of k largest values of Σ. The value of k is smaller than the length of query. The column space of k-rank approximation matrix is a subspace of the column space of A[14].

4.5 Reconstruct word-document matrix

When reconstruct the term-document matrix with formula 4.5.

$$X = U_k \Sigma_k V_k^T$$

In appendix B, we see that reconstructed term-document matrix suggest a rate that a word should be in a document. In the original term-document matrix, the occurrence of ”survey” in document m4 is 1 and that of ”trees” is zero. While in the reconstructed matrix, the value of ”survey” in document m4 is 0.42 and that of ”trees” is 0.66. This suggest that ”survey” should have a lower weight in m4 document and ”trees” should have a higher weight in document m4. This changed relies on the other words in each document, and the ”latent semantic” is obviously reflected. When user searching the document, LSA can retrieve and return the documents even they don’t contain word in query as long as these documents are semantically similar to user’s purpose. With another point of view, value of each column also can be seen as the semantic contribution to the document. Word with hight weight contributes more that the word with lower weight. When comes to the minus value, it can be regarded as a negative contribution to document. Sum of weight of specific words contributions indicate that how much have the document talks about the ”semantic” meaning of query.

4.6 Comparison

Rely on the matrices obtain by SVD, two types of comparison can be applied. The first is comparison of two terms. In matrix M, dot product of two rows represents the similarity of two terms. This dot product can be expressed by AA^T. Since S is diagonal and V is orthonormal is can be verified that $AA^T = US^2U^T$, value in cell i,j can be calculated by dot product of i, j rows in US [12]. Similar to term-term comparison, document-document comparison can be applied in the same way. In matrix A^TA, dot product between two rows represent the similarity to of two documents. Again, $A^TA = VS^2V$ can be generated.

Hence value of i,j in $A^T A$ can be obtained by dot product in VS [12]. For the purpose of fully understanding the relationship between term(s) corpus and document, a query string matrix has to be created according to the original term-document matrix. The value of q is the same form of columns on original term-document matrix. Following formula is utilized to obtain query vector which finally used to compare with document matrix. $Q = q^T U \Sigma^{-1}$ Here, q^T is the transformed vector of q. By this way, query string is mapped to document space. Then it ready to evaluate the similarity between query document and documents.

The cosine similarities between query string and documents are only indicating how close the document related to the query topic, but not how much have been involved about the topic. In order to discover how much or how important of the document about query, we start to concentrate on the reconstructed term-document matrix.

4.7 Limitation

LSA provides an approach to excavate the latent meanings from documents. However, LSA still has some limitations. First, LSA can not handle the word order, which means that the syntactic relation of logic can not be handled. Moreover, this limitation does not affect the process of extracting word and paragraph meaning, but it must still be suspected of resulting incompleteness or likely error on some occasions[24]. Another weakness is event LSA can analyse synonymy in context, LSA can not distinguish polysemy. In default, each occurrence of one word only be treated with same meaning, and in semantic space, each word take on unique position even the word has different meanings.

Chapter 5

Proposed Solution

This chapter will introduce our proposed solution, the XRank algorithm. This chapter also explains the reason why we want include LSA to influence estimation. In details, we demonstrate how we utilize LSA to get document similarity and to reconstruct word matrix that deduces word contribution. In this chapter, we also represent the formula of XRank.

5.1 Proposed solution

5.1.1 Basic algorithm

A variety of information are exhibited on social media includes both content resources and non-content resources [1]. These two types of resources can be variables to estimate online influence. Non-content variables such as user relationships are valuable to online influence. Take Twitter as an example, followers count indicates how many people are interested in influential. At the same time, influential's tweets are easily propagated throughout links between users. Moreover, links from influencer to followers determines the information flow direction which also indicate influence direction. Larger number of followers means higher probabilities of tweet propagation. Another factor in Twitter worthy to mention is the tweet count. Tweet count shows that how many tweets posted by influential. These tweets are the most direct medium that influence other users. Similar to non-content resources, contents are also a valuable factor that can be utilized to evaluate online influence. The approach we proposed is we plan to count both content resources and non-content resources as our variables to evaluate online influence.

We plan to evaluate correlation degree between contents and given topic by utilizing IR methodologies. As we discussed in section 3.2.1 and 3.2.2, both VSM and Naive Bayes classifier have weakness. Thereby we turn to LSA because of several advantages of LSA. The advantages are list as follows:

- LSA can discover latent semantic meaning of document. As we discussed in chapter 4, LSA is capable of finding documents that have close correlation with query. Documents don't contain keywords in query also can matched as long as these documents are literally related to query.
- LSA doesn't need labelled data and any training procedure.

LSA maps vocabulary into document space and compare to get cosine similarity between documents. The document similarity is a very important element used in XRank.

5.1.2 Word contribution

As we mentioned in section 4.5, each row in reconstructed word-document matrix represents the probabilities of one word's appearance in all documents. In each column, words with weight indicates that the document is much more related to the semantic meaning of these words. In contrast, words with lower weight suggests that even these word appear in the document but document is not really about the meaning of these words. Another way of regarding this is summing several cells indicates how much dose the document writes about the ”total” meaning about these ”few words”. And we call it *word contribution*. Word contribution(WC) in reconstructed word-document matrix represents the word ”appearance” or ”not appearance” reasonably. Appendix B is a good example to explain WC. Take document c2 as an example, c2 talk about the user's opinion about ”system response time”, since only the word occurrence twice will be analysed. We can view that c2 is more talk about the ”computer system response time”. In reconstructed term-document matrix, the highest values belong to system, user, response and time which are 1.23, 0.84, 0.58,0.58, separately.

In reconstructed word matrix,WC value can be zero, negative and positive. The approach of processing word contribution for document is adding each word contribution together. And the total word contribution can be zero negative and positive as well. Zero contribution value can be view as there is no contribution to the document in total. Negative contribution value indicates that the document are not very related to the topic. It is easy to treat the positive value as an affirmative contribution of vocabulary word to document. Larger WC value means that document is more related to the topic than the lower words of contribution. For example, as shown in Appendix B, assume that we want to search document about ”computer response time”, we can see that the word contribution for c1, c2 c3 c4 and c5 are 0.47, 1.72, 1.3, 1.5 0.79, separately. At the same time word contribution for m1, m2, m3 and m4 are 0.02, 0.07, 0.11 and 0.27. The difference is significant between two types of documents except document m4. Document m4 is not really about ”computer response time”, we think that because of the existence of ”survey”. That will not be effecting the LSA understands the document, since we are going to combine this value with cosine similarity to form a new value called *document impact value*.

5.1.3 Document impact value

In chapter 4, it has been discussed that cosine similarity between document and vocabulary expresses the affinity between document and vocabulary, and WC is on behalf of how much the document has involved in one topic. We can get the document impact value from document similarity and WC. Then the document impact(DI) value represents not only how close the document related to specific topic, but also how much the document has written about the topic. Section 5.2 will describe how to obtain document impact value.

5.1.4 Retweet and mentions on Twitter

In Twitter, retweet is an experience of sharing interesting tweets, links from other users. A retweet is often starting with ”RT”and ”@username” followed by interesting content. For example, user Jackson with user name of ”jackson” published a tweet ”New iphone 5 will be release this September.”, and user George’s retweet would be like ”RT @jackson New iphone 5 will be release this September.”. Mentions are identified by tweets containing ”@username”, excluding retweet. Mentions start with ”@username” is only to tweet replier and ”@username” in the middle of tweet are broadcasting to all followers. The count of retweet or mentions suggestion how many users are influenced. Vast amount of retweet and mentions reflects that user have high probabilities of chances of transmitting influence. Therefore we take retweet as an important variable in XRank.

5.2 Prototype design

The prototype is revealed in Figure 5.1.

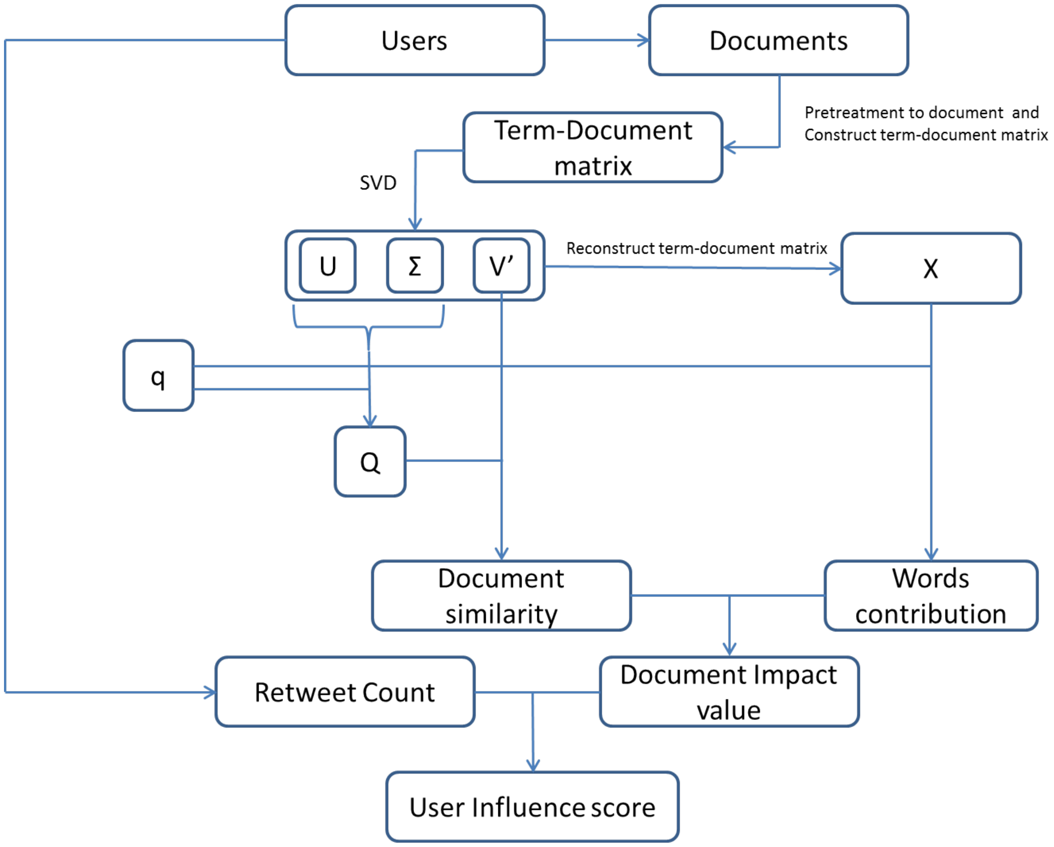


Figure 5.1: Prototype design of XRank algorithm. q is the query string that contains retrieval keyword Q is the query vector in document space from q. X is the reconstructed term-document matrix which derived from U, Σ and V^T.

In this proposed solution, we assumed that the document correlation with given topic is a significant variable of determining online influence. Hence, we plan to collect tweets for users and to combine amounts of tweet together to be a document. Document pretreatment removes useless content such as links and "@username" from documents. After documents are cleaned and they are ready to be transformed into term-document matrix. The method of creating term-document matrix has already been discussed in chapter 4.2.

Next critical step is applying SVD to the term-document matrix. Three new matrices are generated which are U, Σ, V^T. Based on these three matrices, we are able to reconstruct term-document matrix. Combining with created vocabulary,we can find word contribution for each documents with given vocabulary. Additionally, vocabulary is also mapped in to document space. By calculating the cosine similarity between query vector (Q) and all document vector in matrix V, we can fin the similarity between document vectors.

DocSim is represented by cosine value between two vectors. Therefore the similarity value ranges from-1 to 1. Value 1 means that document is semantically similar to vocabulary, while -1 meaning totally different. This value is denoted as $CS(Q, D_i)$. We would like to normalize this value into document similarity with a positive with formula 5.1.

$$DocSim = \frac{1}{1 + cos^{-1}(CS(Q, D_i))} \tag{5.1}$$

$cos^{-1}(CS)$ is the radian angle between Q and document vector in document space. Value of $cos^{-1}(CS)$ is from 0 to π. Adding 1 to $cos^{-1}(CS)$ is to avoid zero-division. $DocSim$ is a function of $cos^{-1}(CS)$. Final value would be from $\frac{1}{1+pi}$ to 1. If the angle is 0, $DocSim$ will be 1. While if two vector are opposite, $DocSim$ will be $\frac{1}{1+pi} < 1$.

Every WC value can be negative, zero and positive. Only positive value means contribution to document, negative value does not. XRank algorithm only counts positive contribution. wi is the word in query string and con_{wi} is a positive value. Negative WC values are ignored.

$$PWC_{di} = con_{w1} + con_{w2} + ... + con_{wn} \tag{5.2}$$

Based on the $DocSim$ and PWC, we can obtain DI value with formula 5.3

$$DI_{di} = DocSim(Q, D_i) \times PWC_{di} \tag{5.3}$$

The way of obtaining tweet count for each user is counting how many times for all tweets in test dataset is retweeted, then add them together. We noticed that the test dataset is small even it contains hundreds of thousand tweets. And we can not get how many retweet for each user in total, thereby we would like to use *retweet rate* to evaluate the user's capability of enabling the rest to retweet. TC_{ui} is the tweet number for each for user i in test dataset, and RTC_{ui} is the count of how many retweet for the tweet in test dataset. We found that for each tweet, the number of retweet is often lower than 100. For cases of retweet value larger than 100, Twiiter API only provide ”100+”. Therefore, as an expedient way we count ”100+” as 100 for all.

$$RTR_{ui} = RC_{ui}/TC_{ui} \tag{5.4}$$

In order to derive user influence value, we utilize formula 5.5. We choose 3 as the base of logarithm is due to the empirical practice. Since have to give each of parameter with a proper weight and with many tests, we find XRank result is good when we choose 3 as the base of logarithm. The reason we add 3 to RTR_{ui} is avoiding negative value when RTR_{ui} is smaller than 3.

$$UI = Log_3(RTR_{ui} + 3) \times DI_{di} \tag{5.5}$$

Finally, we have our final formula:

$$UI_i = Log_3((RC_{ui}/TC_{ui}) + 3) \times \frac{1}{1 + cos^{-1}(CS(Q, D_i))} \times PWC \tag{5.6}$$

UI_i represents User Influence for user *i*, $RC_{u,i}$ represents retweet count for user *i* in test dataset, $TC_{u,i}$ represent tweet count for user *i*, $CS(Q, D_i)$ represents cosine similarity between query and document i that is on behalf of user *i*, PWC represents positive word contribution of words of document i.

5.3 Correlation evaluation methodology

5.3.1 Spearman ρ correlation measures

XRank algorithm will deduce a rank for all users in test dataset. We want to compare XRank result with Klout Score rank. We select Spearman ρ correlation coefficient to estimate correlation. Spearman ρ correlation coefficient can evaluate relationship between two variables by utilizing using a monotonic function.

Spearman ρ correlation coefficient is often denoted with Greek letter ρ and it is used to measure association between two ranks. Perfect value of Spearman correlation coefficient is -1 or 1 if variables in two ranks are monotone function with each other. If the Spearman correlation coefficient is close to 0, the two ranks are independent. Spearman correlation coefficient is calculated by the following formula if tied ranks exist, this formula is described in [27]. Assume that there are tow vectors X and Y with the same dimension, the Spearman rho correlation can be calculated with the following formula:

$$\rho = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 (y_i - \overline{y})^2}} \tag{5.7}$$

here, x$_i$ is the i-th value in rank X, and $\overline{x}$ is the average value of all x in rank X.

Chapter 6

Experiment

We organize this chapter with button to top pattern. First we describe how we collected the Twitter data and pretreated the documents. We then suggest four test cases where we apply XRank algorithm. Afterwards, we exhibit the test results for all test cases. Based on these results, we explain the results and make discussions. At the end of this chapter, we summarize the test result and the discussions.

6.1 Experiment settings

6.1.1 Dataset characteristeics

In this experiment, we would like to apply XRank algorithm to Twitter data. Twitter provide a Twitter API[*] that exhibits plenty of interfaces for developers to develop Twitter relevant applications. There are a variety of APIs that can be use to manipulate Twitter data with or without authentication. To gather data to our experiment , we only need to use the Twitter search API. By utilizing the search API, we get both user resources and timeline resources on Twitter. Either we can select users randomly, or we can acquire users from the Twitter suggestion list. Finally we decided to get user from suggestion list. Twitter API has a suggestion API [†] that can get famous users that range from art and fashion to health and technology. For our purpose, we collect 42 users from technology area and 40 users from art-design field.

A non-ignorable fact on social media is that user number is growing everyday and

[*] http://dev.twitter.com/doc [†] http://api.twitter.com/1/users/suggestions.xml

contents are also increasing remarkably. Take Twitter as an example, 572,000 new accounts are registered only on March 12, 2011. In 2010, increase rate of mobile users is to 182% . There 177 million tweets sent on March 11, 2011. In February, 2011, 140 million tweets were sent by people everyday [37]. Meanwhile, another matter of fact is that only small part of all users are contributing a lot. Some users never post any tweet after registering. A research report release by Barracuda Labs reveals that 34% of Twitter users have no tweets since they created an account and 73% users have less than 10 tweets[7]. These statistics data shows that a large amount of users are content consumers. In our test, we want to make sure that tested users are have enough tweets. Users in Twitter suggestion list are active users and have a lot contribution to content. However, because of technique reasons, some users in the list don't fulfil our requirement. Finally we have 83 users to test.

Twitter suggestion list contains 19 topics[*]. From these' 19 topics, we select *technology* and *art/design* topic. Because we want to test XRank algorithm's performance when Applying XRank with different topic vocabularies. The test case details are described in section 6.2.2.

Twitter API provide a very convenient approach for developers to retrieve and modify twitter data. However, it has limitations for anonymous developers. For instance, the timeline search API [†] only return 3200 tweets for each user. As this reason, in the test dataset, tweet count for each user is not more than 3200. In Chpater 5, we discussed the step of involving LSA into XRank. We proposed to utilize a document to represent a user for content analysing. Therefore, we merge a certain amount of tweets as a single document. The amount of tweets is determined by test case which will be demonstrated in section 6.2.

6.1.2 Document pretreatment

XRank is designed to read human language that consists of words. Therefore non readable words should be removed from each individual tweet. In many cases, Original tweet consists of some irrelevant contents. Following text shows two typical tweets:

> Tweet 1: People may not love the Kindle - but they love the Kindle package design: http://tinyurl.com/2q2ngq

[*] The topics consist of Art & Design, Books, Business, Charity, Entertainment, Family, Fashion, Food & Drink, Funny, Government, Health, Music, News, Science, Sports, Staff Picks, Technology, Travel and Twitter [†] http://api.twitter.com/version/statuses/usertimeline.xml

> Tweet 2: @dsifry but what I thinking @plasticbagUK is betitng is that disadvantages will outweigh advantages of books. date is debatable, trend clear

Tweet 1 contains a typical tiny link *http://tinyurl.com/2q2ngq*. Tiny link is a short aliases for long Uniform Resource Locator(URL) for redirection. Such links are hash indexes in the server of Tinyurl*. Therefore tiny links don't have plain meaning and they are useless to semantic analysis. Hence such links will be removed before this tweet is assembled to document. Tweet 2 contains reply symbol ”@dsifry” at the beginning. It also contains mentions symbol like ”@plasticbagUK” in the middle of the tweet. Although those symbols are meaningful. However, we don't need to analyse replies or mentions for XRank. Hence those symbols also would be removed. Other symbols like ”-” and punctuations will be removed. Additionally, numbers in tweet are often refereed as time or quantity. They have to be removed as well. But abbreviations such as ”G2” or ”3G” will be kept since lots of electronic products are name in that way.

6.1.3 Vocabulary creation

XRank is an algorithm that estimates online influence based on given topic. To specify a topic, we need to create vocabulary. This vocabulary consist of keywords belong to the same topic. It is not necessary to consider word order or logic between these words. As discussed in chapter 4, LSA can not handle the word order in documents. Therefore, artificial vocabulary will be still viewed as a document. A feasible way of creating vocabulary can be divided into following steps:

- Collect text content such as tweets, news or articles in specific topic, and statistic the count for each word.
- Set a threshold for the for previous statistics. Word occrence larger than threshold will be selected to a new collection.
- Select keywords which are related to a given topic.

To create a vocabulary related to technology topic. We gathered 42 documents which consist of tweets from technology topic. Then we calculate occurrence for each word. Afterword, threshold is set to 5. That means that word occurrence more than five times are collected. The reason we set threshold is to ease the burden of selecting keywords. In

* http://tinyurl.com/

this case, the collection contains more than 10,000 words even we set to 5. The larger the collection is, the more time takes to select keyword. However, if threshold is too large, some keywords are probably filter. So the suggestion is that choose a proper threshold depending on dataset. If dataset is large, then consider to use a high threshold and vice versa. Part of word collection we crated is like this:

> *billboard bottom creative fog **Facebook** Keith Instant Calls Thing losing Think First Subject citizens WaltMossberg Valentine slightly raised Knowledge Hybrid Management magazines raises shoots support Hand **Device** techmeme launching offer Carbon inside devices Francisco floor Trac LivingSocial developers Empire Realtime Buffett Entrepreneurship **smartphone** downloading subway Tab team da3mon **Tablet** sign shirts Techmeme Videogame myTouch **Twitter** brilliant ShareablesThe MS **iPhone** Stealing love prefer Enough fake red August working printing Newsweek Federal Symphony unveiled allowed monitoring winter Who **googlebooks** elephant extensions applications **Cisco** Catching truck pulled Barne **Verizon** bagels recipients smarter nation subscribers Experiment Broadband Livy Live **Honeycomb** Satellite Elizabeth MichiganLeopard Job investigation Ghostbusters internet Article Jon Places million training saving **HD2 Groupon** philanthropy relationship **TED** Long theater Cancer **Telecom** marriage Click HQ Movie graffiti Checkout Food Brings internal play BBC M chrissyteigen plan accepting Fahey **Google Cloud** writer Labs failed factor Celebrate Spice Prime banned **Wifi** sunny preparing*

There are still many irrelevant words in the collection such as *August, Live,Places*. Those words are very common and they can be appearing in all types of documents. Therefore we need to filter common words and select keywords by ourselves. In the collection above, the words marked as bold face are technology keywords. We can see that there are several types of keywords. First type is company name or organization name such as *Google, Facebook, Twitter, Groupon, Verizon, TED*. The second type is business product name such as *HD2*[*]*, googlebooks, iPhone*. Another type is common words related to technology such as *device, smartphone*. And the forth type of keywords is polysemy words such as *Cloud* which probably refers to cloud computing or to meteorology cloud. Since cloud computing is very popular topic in technology area, we want to keep it in our vocabulary. From the collection above, we obtained the vocabulary for technology topic as follows:

[*] HD2 is smartphone producted by HTC

> *Ericsson google Slate LG iphone API APIs Code Twitter PS3 Android developers Device Incredible Tablet applications Cisco Verizon Satellite Honeycomb phone technology 3GS Inspire Xbox legend Panasonic htc nano diamond2 ted telecom cloud labs wifi gmail chip mac vodafone buzz translate pro2 xperia wwdc webcast ipad pc geo apple hardware crack adobe webos kindle smartphone digital carriers wildfire Social Media network Microsoft cameras gallery computer Sense spotify voip ios nexus dropbox tablets 3DS blackberry adsense podcast antenna fcc SD IBM mobile itunes battery skype mwc a4 youtube facetime ipod yahoo zune gsm googleio xoom chrome Netbook Amazon Canon Hulu keyboard MacBook NFC jailbreak LTE Disk Screen 3D Motorola 4G G2 G1 McAfee Hero laptop hp HD wireless Flyer Opera Samsung linux Bluetooth iMac 16GB 32G iPhone4 sony GPS Zynga ARM*

Beside technology vocabulary, we also created a vocabulary for art/design topic in the same way. By analysing tweets which are posted by users who are in art suggestion list, we have the word collection as follows:

> *House dirty Close* ***music*** *Obsession Factor watches Hot Production* ***design*** *watched cream Read extract Simpsons Palin Scott Lady teen door company art Cheese keeping science installing learn marthastewart Conference suggestions Silver found Sounds HDR favorites* ***historian*** *number Kagan Fry guess guest jet introduction Guardian relationship interviewed mural stairs Shuffle Click* ***Twitter*** *Movie fights graffiti sell* ***Graphics*** *Club Brings Akzidenz internal play brooklyn plan* ***Google*** *cover artistic Portfolio Auction gold Had session Has Holzer writes writer* ***Facebook*** *Penguin Prime Masters sunny obscure creator Fox photographs Glaser Fog sea Song exchange fantastic Redesign Against Server death Koons Annie interface improved Louvre Chip connection* ***amazing*** *Todd electrician loan* ***photographs*** *readers admission Arial danke eager parents Mann Marina submissions surprised Both doors couple calreid Ask projects continue stylish Summers composer Niemann pals Andy sight* ***print*** *curious Friend Frank Look Pace majo Shaw* ***canvas*** *hometown recreate* ***illustrated*** *gossip* ***Designer*** *young send Glass* ***Designed*** *torture continues animals Fixed magic*

In art/design word collection, company names such as *Facebook and Twitter* appear in document as well. This probably these two websites are the most popular social media platform. And people would like to involve topics related to these two websites. However, we believe that such company names are not relevant to art/design. But organization name such as *Louvre* are considered as an art related keyword. Because as is well known, Museum Louvre is one of the most famous art museum in the world. Another type is graph relevant words like *photographs and images*. These words are items that relevant to design materials. Words like *font, Arial, theme, pixels and image* are more interrelated to website User Interface(UI) design. Moreover, adjectives such as *amazing, elegant, impressive* are kept because those words are probably related to art/design works.

> *music art design beauty webfonts Beautiful feeling petapixel pixel theme conceptual life magazines wordpress panels choice awesome artist vision IKEA Vienna favorite gorgeous sense imagine Redesign interface view stylist stylish Designer Designed magic Flash designs scene Auction Advertising advertise color patterns inspiration Photograph musical architectural pink typographica brown bold paintings photoartgallery cartoonist Graphics retrospective dresser museum Geneva fantastic amazing Sculpture sunflower carving photographyelf Studio designblahg images print paint letterpress Classic font solid jewelry pet map cartoonists Imprint inspiring photographs romantic Louvre elegant tnycloseread Ivy prints picture impressive artworks wood cabinet canvas sculptural Arial style grace essential showcase Light necklace Arts Landscape vintage Avatar concerts Writer whitney*

It is worth to mention that XRank will not distinguish upper-case and lower-case. Which means ”Google”, ”google” or ”GOOGLE” will be considered as the same word as ”google”. When creating term-document matrix establishment and vocabulary vector, all words are lower-cased to standard form. Because case of letter is not important in index items. Many IR system convert items to either upper case or lower case[13].

6.2 Test cases

6.2.1 Test case A: Test XRank algorithm on vocabularies with different sizes

In test case A, we choose 42 users related to technology topic. For each user, we search 500 tweets from Twitter and combine them into one document. In previous section, we described how to create technology related vocabulary. Then we construct three types of vocabularies. Full size vocabulary contains 130 keyword. By removing 66 keyword, we get a half size vocabulary contains 64 keywords. To create a small size vocabulary, we selected nine words from full size vocabulary. All those vocabularies can be found in Appendix H.

Table 6.1: Parameters in test case A. User Quantity is the user number in this test case. User Topic refers as field of user relates to.

User Quantity	**User Topic**	**Document Size**	**Vocabulary type**	**Vocabulary Size**		
42	Technology	500 tweets/doc	Technology	Nine words 9	Half size 64	Full size 130

In chapter 5, we described how we include LSA into XRank. LSA compares document with a fake document which is a artificial vocabulary. Hence an indispensable element in XRank is the pre-constructed vocabulary. We want to figure out how vocabulary size would affect XRank result. In order to eliminate other interference, the users we select are all from technology topic. Additionally, the vocabulary type is also technology. And 500 Tweets for each document is a applicable size for testing.

We will keep user quantity, user topic, document size and vocabulary type the same. By changing the vocabulary size, we apply XRank algorithm to test data and will get three results. We will compare these result to examine how vocabulary sizes affect XRank result.

6.2.2 Test case B: Test XRank algorithm on vocabularies with different content

This test is applying XRank algorithm to 83 users from both technology and art/design topic. Among them, 42 users from technology topic and 41 users from art/design topic. Then two different types of vocabularies will be created. One vocabulary is related to technology topic. Another is related to art/design topic. They can be found in Appendix H and Appendix I separately.

Table 6.2: Parameters in test case B. When testing, 42 users from technology area and 41 users from art/design users are mixed estimated.

User Quantity	**User Topic**	**Document Size**	**Vocabulary type**		**Vocabulary Size**	
42	Technology	200 tweets/doc	Technology	Art	**Technology**	**Art**
41	Art and Design				130	106

As we mentioned in section 6.2.1, XRank has a tight relationship with vocabulary. In section 6.2.1, we designed test case A to see how vocabulary size would affect XRank. At the same time, it is also very interesting to examine how vocabulary topic affect XRank result. So we select users from both technology and are/design topic. Then we create two vocabularies regards two different topics separately. Each document consists of 200 tweets. Considering the dimension of term-document matrix from 83 documents would be very large and our LSA library can not handle large dimension matrix, 200 tweets is empirical choice.

In the first round test, we will test XRank algorithm with technology vocabulary. The result will show the online influence related to technology topic. In the second round test, XRank algorithm uses art/design vocabulary. As a premise knowledge, we have known that what topic of each user related to. Then we will check the result to see if technology users have high rank in the first round test and low rank in the second round test. And vice versa. If the answer is yes, we can say that XRank has a good capability of ranking online influence based on topic. In contrary, then XRank is fail to rank online influence based on topic.

6.2.3 Test case C: Test XRank algorithm based on variance of dataset size

This test will apply XRank to 42 users from technology topic as the same we used in test case A. We will use full size technology vocabulary to eliminate the reduce impact from other variables. We will create three datasets to derive rank from XRank. Each dataset contains corresponding 42 documents that every document consists of 200 tweets, 500 tweets and 800 tweets separately.

Table 6.3: Parameters in test case C. There are three independent tests in test case C. Each time will use a different document size to test XRank algorithm. The rest parameters such as users and vocabulary are the same.

User Quantity	**User Topic**	**Document Size**			**vocabulary type**	**vocabulary size**
42	Technology	Small	Middium	Large	technolgy	130
		200	500	800		

Since we include content analysis in XRank algorithm, one of the most significant missions is gathering content. In this test, we collected a certain amount of tweets from Twitter. From each users in test, we merge a number of tweets to form a document. Then we face the challenge of figuring out how many tweets should be selected. From the results we will know how XRank results change according to the varying of dataset size change. This will provide a clue to find out a proper number for creating document.

6.2.4 Test case D: Comparison between result derived from XRank and Klout Score

In test case D, we use the exactly the same parameters as in test case B. Additionally, we obtain Klout Score for these 83 users by utilizing Klout Score developer API *. Then make a comparison between XRank results and Klout Score.

6.3 Results and discussion

6.3.1 Result A

After applying the XRank algorithm to 42 users with three types of vocabularies, we have result A. Result A contains three XRank results. These results are exhibited in Appendix C. Among 42 users related to technology topic, we select four users who are *google, mashable, wired* and *chadfowler* to explain the result. Results for these four users are represented in table 6.6 to 6.8.

In the XRank result based on the full technology vocabulary, the users *google* and *mashable* are both top rank users. User *google* has a low document similarity(0.41). This means that user *google* talks covers part of topics in vocabulary. Probably more about it's products. In order to prove our conjecture, we investigated user *google*'s tweets. Most of the tweets talk about google products such as google earth, google buzz, andriod, google code, etc.. Meanwhile this user have a very high WC value(443). High WC value indicates that some words in vocabulary are mentioned a lot and these words have tight relation to technology topic. User *google*'s word contribution are shown in table 6.4. From table 6.4, we see that word ”google” contributes half of total contribution. And ”youtube” ”gmail” ”mobile” are contributing a lot as well. This means user *google* talks a lot about its product such as ”google earth, google buzz”, therefore word ”google” contributes a lot. On the other hand, words like ”itunes”, ”imac”, ”adobe” have very low contribution. Such low contribution words indicate either these words rarely appear in google's tweets or these

* http://developer.klout.com/api_gallery

words don't related to google's main topic even they arise a certain amount of times.

Table 6.4: Word contribution for user *google*. This result is arrived from XRank algorithm with 500 tweets per document and with full size technology vocabulary. This table only shows 16 words that contribute a lot on the left two columns and 16 words in vocabulary but contribute very little on the right two columns.

Word	**Contribution**	**Word**	**Contribution**
google	222	apis	1
youtube	54	hd	1
gmail	24	sense	1
mobile	15	hardware	1
twitter	11	developers	1
labs	10	microsoft	1
cloud	10	itunes	2.17E-15
googleio	9	wifi	2.16E-15
chrome	9	cisco	1.76E-15
android	8	adobe	1.01E-15
api	6	imac	2.40E-16
code	5	a4	6.71E-16
3d	5	hulu	6.80E-16
technology	4	geo	3.30E-16
applications	4	jailbreak	1.48E-16
media	4	sd	1.47E-16

Table 6.5 shows the part of word contribution for user *mashable*. Since the most contributory words are *"social" "media" "ipad" "twitter"* and *"mobile"* , we deduce that user *mashable* talks about social media network and mobile devices.

In table 6.8, due to low document similarity and high word contribution, user *google* has a middle document impact (182.9) Among all users, highest document impact is 358.7 and 7.5 is lowest value. In contrast, user *mashable* has a very high document similarity which indicates what this user writing is very close to vocabulary. Another user is worth to mention is *wired*. This user has a low document similarity (0.4) and medium word contribution(300) with a middle retweet rate (33.95). Finally this user still gets a high XRank result (393.79). User *chadfowler* has a low document impact (8.48) due to low document similarity(0.42) and low word contribution(20). Moreover, the retweet rate is very low(0.501) as well. Therefore user *chadfowler* has a very low XRank result(9.68).

From table 6.6 to 6.8, we see that DocSim doesn't deduce rigorously with decreasing of vocabulary. We also notice that user *google* has a higher DocSim when XRank utilizing nine words vocabulary than XRank utilizing full/half size vocabulary. A possible explanation is that full/half size vocabulary contains many keywords that user *google* doesn't write about. By removing most of irrelevant keywords, relevant words are kept. Hence document

Table 6.5: Word contribution for user *mashable*. This result is also arrived from XRank algorithm with 500 tweets per document and with full size technology vocabulary. This table shows 16 words that contribute a lot on the left two columns and 16 words in vocabulary but contribute very little on the right two columns.

Word	Contribution	Word	Contribution
social	60	incredible	1.72E-15
media	39	satellite	1.64E-15
ipad	22	laptop	1.59E-15
twitter	17	panasonic	1.40E-15
mobile	17	facetime	1.27E-15
android	16	jailbreak	1.23E-15
google	14	crack	1.05E-15
chrome	14	opera	8.11E-16
iphone	13	tablets	7.54E-16
youtube	11	lg	7.52E-16
network	9	telecom	5.58E-16
tablet	5	webos	4.94E-16
cloud	5	geo	4.73E-16
amazon	4	3gs	4.35E-16
device	4	sd	1.58E-16
4g	4	imac	1.41E-16

Table 6.6: XRank result based on nine words in technology vocabulary. From table 6.6 to table 6.14, DocSim is the document similarity which is processed cosine similarity, WC is word contribution, DI is document impact score. RC refers to retweet count, TC represents tweet count, RTR stands for retweet rate.

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
1	google	0.799	222	177.452	84295	2158	39.062	604.00
8	mashable	0.456	17	7.749	186144	3200	58.17	29.0
40	wired	0.761	19	14.460	106202	3128	33.952	47.5
36	chadfowler	0.856	2	1.712	1456	2901	0.502	1.95

has higher similarity to remaining word in vocabulary. From right of Figure 6.1, we can see trend of DocSim for the four users clearly. Except user *google*, the rest of three's document similarities are decreasing with the growing of vocabulary size.

Figure 6.1 shows DocSim trends based on vocabulary size change. Left figure shows average DocSim value based on vocabulary size change. Average value decrease from 0.668 to 0.656 then to 0.623. This change is not remarkably and every value is close to median(0.621). Right part of Figure 6.1 is the DocSim change for user *mashable, google, wired and chadfowler*. Some users' DocSim are becoming larger and some document similarities are lower down.

Figure 6.2 demonstrates the WC trends according to the variance of vocabulary size.

Table 6.7: XRank result based on half size technology vocabulary

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
8	mashable	0.89	232	206.13	186144	3200	58.17	771.82
1	google	0.43	337	145.30	84295	2158	39.062	494.5
40	wired	0.42	187	78.50	106202	3128	33.95	257.9
36	chadfowler	0.45	16	7.15	1456	2901	0.502	8.2

Table 6.8: Rank result based on full technology vocabulary.

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
8	mashable	0.98	302	295.41	186144	3200	58.17	1106.1
1	google	0.41	443	182.86	84295	2158	39.062	622.4
40	wired	0.40	244	98.17	106202	3128	33.952	322.6
36	chadfowler	0.43	21	8.97	1456	2901	0.502	10.2

Overall trend is word contribution decreases as size of vocabulary size. As more keywords in vocabulary, the difference between word contributions are more distinct. From Figure 6.2, we can see that WC increase very significantly for every user. Word contribution has a very tight relationship with numbers of keywords in vocabulary. Decreasing word number directly leads to reduction of contributory word and vice versa.

Figure 6.3 describes average value of similarity result for three conditions. Average XRank result increases when vocabulary size grows up. However, when vocabulary from nine words to half size, the rank order changes. But when vocabulary size changes from half size to full size, the order doesn't change.

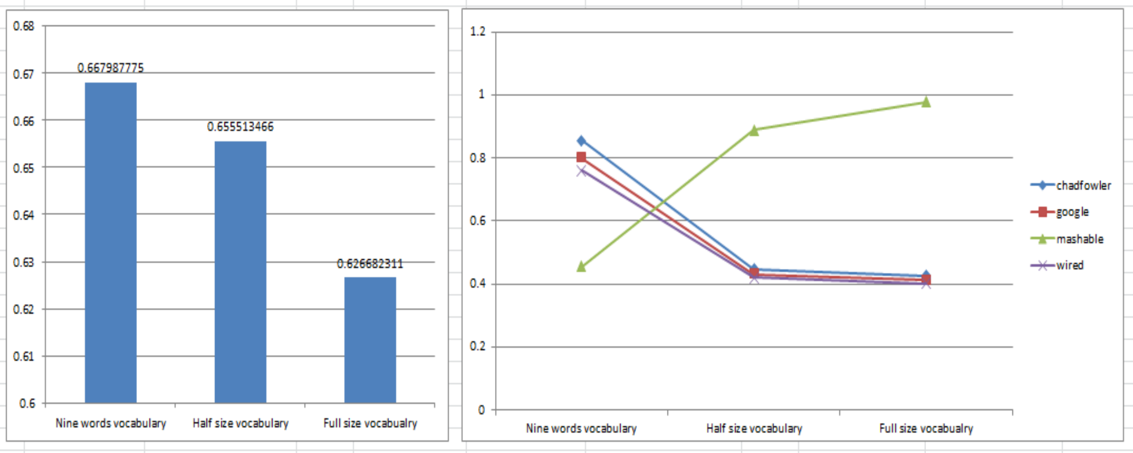


Figure 6.1: DocSim trends based on vocabulary size change, from left to right are full size vocabulary, half size vocabulary and vocabulary with nine words. Column figure on the left shows average value of all document similarities for each precondition. Line figure on the right shows document similarity change for four users.

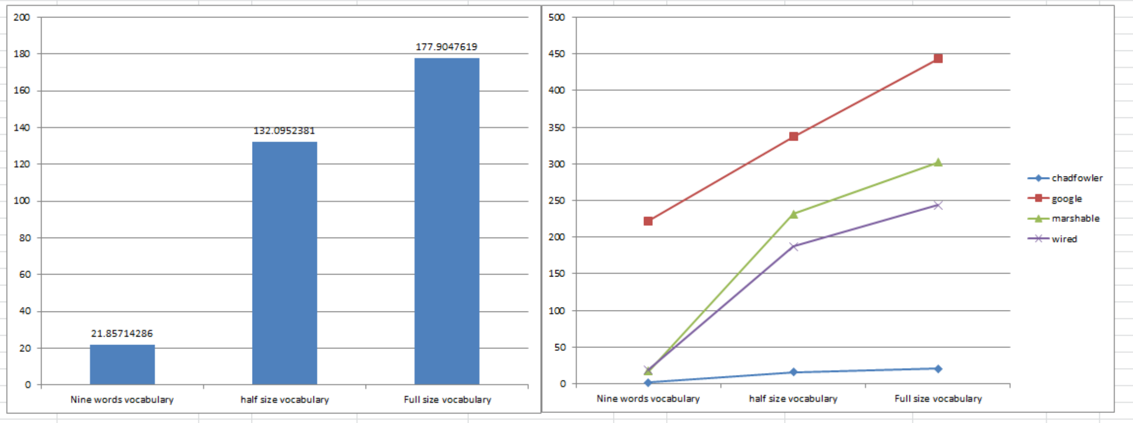


Figure 6.2: WC trends based on vocabulary size change

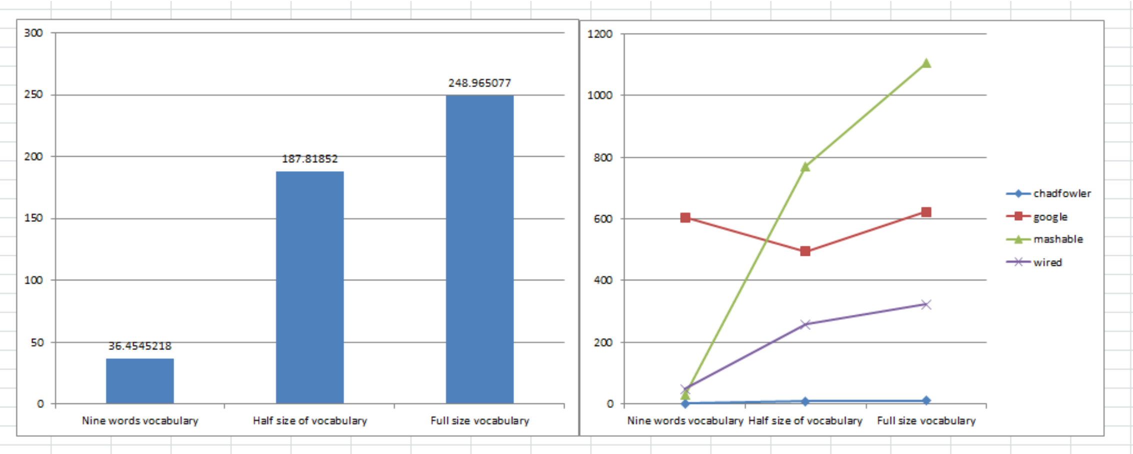


Figure 6.3: XRank trends based on vocabulary size change

6.3.2 Result B

Table 6.9 shows the XRank result base on technology vocabulary and art/design vocabulary. XRank(T) indicates the column is the rank result base on full size Technology vocabulary, XRank(A/D) represents that column is the XRank result based on full size Art/Design vocabulary. User *google*, *gadgetlab*, *TechCrunch* and *mashable* are top four rank users in XRank(T), user *printmag*, *AIGDesign*, *designmilk* and *artinfodotcom* are top rank users in XRank(A/D). The top four users in XRank(T) have a lower rank in XRank(A/D) and vice versa.

Table 6.9: XRank result based on on both full size technology vocabulary and art/design vocabulary.

ID	Name	XRank(T)	XRank(A/D)
49	gadgetlab	337.4	11.87
27	mashable	314.7	24.8
47	TechCrunch	312.0	18.4
4	htc	254.7	15.0
29	printmag	29.8	221.4
37	designmilk	10.4	216.2
9	AIGdesign	12.9	211.00
7	Tate	12.2	179.8

After we applied XRank to users from both technology and art/design topic with full size technology vocabulary, we have the result shown in table 6.10. Full result can be found in Appendix **??**. Users have high rank in XRank(T) all have high document similarity and high word contribution. Hence, they have high document impact score. Users related to art/design topic has very low document similarity(less than 0.4) and low word contribution(less than 55). These two reasons mainly leads to low XRank result.

Table 6.10: XRank result based on full technology vocabulary, 83 users from both technology and art/design topic

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
49	gadgetlab	0.96	162	155.65	15554	1988	7.82	337.4
27	mashable	0.75	112	84.03	186144	3200	58.17	314.7
47	TechCrunch	0.83	111	91.97	123122	3195	38.54	312.0
4	htc	0.82	149	121.75	186144	3200	58.17	254.7
29	printmag	0.40	43	17.04	8619	2248	3.83	29.8
37	designmilk	0.36	15	5.32	17389	3142	5.53	10.4
9	AIGAdesign	0.40	17	6.78	7889	1554	5.08	12.9
7	Tate	0.35	15	5.34	13483	2987	4.51	12.2

Table 6.11 shows part of XRank result on 83 users with art/design related vocabulary.

Table 6.11: XRank result based on full art/design vocabulary, 83 users from both technology and art/design topic

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
29	printmag	0.93	136	126.56	8619	2248	3.83	221.4
37	designmilk	0.82	135	110.78	17389	3142	5.53	216.2
9	AIGAdesign	0.92	121	110.91	7889	1554	5.08	210.9
7	Tate	0.83	95	78.71	13483	2987	4.51	179.8
49	gadgetlab	0.40	14	5.48	15554	1988	7.82	11.87
27	mashable	0.44	15	6.62	186144	3200	58.17	24.8
47	TechCrunch	0.42	13	5.44	123122	3195	38.54	18.45
4	htc	0.42	17	7.16	10698	1537	6.96	15.0

Appendix H) contains full result. Very similar with previous XRank(T) result, users related to art/design topic have higher document similarity and higher word contribution than that of users who related to technology topic. Even these users don't have high retweet rate as technology users have, they still can get a high XRank result. A very interesting user is *google*. This user has a high similarity to art/design topic and low similarity to technology topic even we know this user should be more related to technology topic.

This test shows that XRank result is depending on vocabulary type to a great extent. Which means by creating different types of vocabulary, we can rank users for specific topic.

6.3.3 Result C

We select the same four users as we did in Result A. From table 6.12, we can see that users have either high word contribution(*google*) or high document similarity (*mashable*) or high retweet rate (*mashable*) will get high XRank result. If parameters includes document similarity, word contribution and retweet rate are low, then user would not get a high rank(*chadfowler*).

Table 6.12: XRank result based on full technology vocabulary, 42 users from both technology topic, 200 tweets for each user

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
8	mashable	0.82	112	91.43	186144	3200	58.17	342.4
1	google	0.43	167	71.00	84295	2158	39.06	241.6
40	wired	0.37	83	31.08	106202	3128	33.95	102.1
36	chadfowler	0.41	7	2.89	1456	2901	0.50	3.3

In Figure 6.4, the average document similarity changes from 0.59 to 0.63 then to 0.62.

Table 6.13: XRank result based on half technology vocabulary, 42 users from both technology topic, 500 tweets for each user

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
8	mashable	0.98	302	295.4	186144	3200	58.17	1106.1
1	google	0.42	443	182.86	84295	2158	39.06	622.4
40	wired	0.41	244	98.17	106202	3128	33.95	322.6
36	chadfowler	0.43	21	8.97	1456	2901	0.50	10.2

Table 6.14: XRank result based on full technology vocabulary, 42 users from both technology topic, 800 tweets for each user

ID	Name	DocSim	WC	DI	RC	TC	RTR	XRank
8	mashable	0.92	467	431.20	186144	3200	58.17	1614.6
1	google	0.41	692	286.22	84295	2158	39.06	974.2
40	wired	0.41	346	143.28	106202	3128	33.95	470.8
36	chadfowler	0.45	40	17.94	1456	2901	0.50	20.5

All these values are just floating around median. We are more interested in single value change. Figure 6.4 shows average document similarity trend as dataset changes. For users who have low document similarity, dataset size change doesn't affect similarity remarkably. However, users have high document similarity are sensitive to dataset size. The reason probably that users have high document similarity usually talk large rang of topics. When words related to one specific topic are growing, the similarity probably increases. While if word related to one specific topic are decreasing, the similarity still will be lower down. From the right part of Figure 6.4, we can see that, the three document similarities doesn't changes much as the document size increasing from 200 tweets per doc to 800 tweets per document. This indicates that LSA doesn't need too many tweet or very big size document to quarry similarities.

However, document size will affect word contribution significantly. Word contribution is always increasing with growing of dataset as shown in Figure 6.5. Since word contribution reflects how much the topic has been talked. In most cases, small size dataset would have fewer information as large dataset, hence less topic will be mentioned in small size dataset.

Figure 6.6 describes XRank result based on dataset change. Apparently, large dataset leads large XRank result and vice versa. This indicates that XRank depends on dataset and large dataset would expand value range.

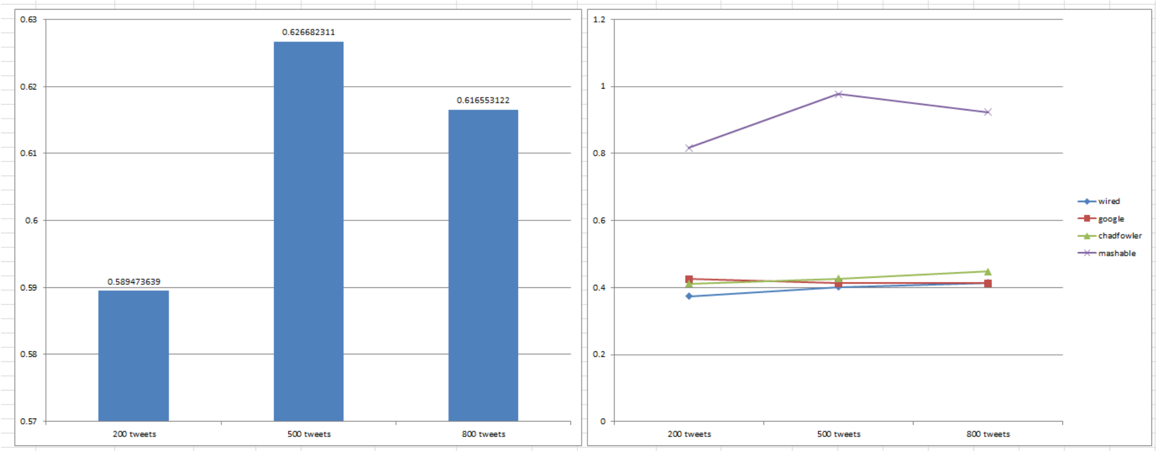


Figure 6.4: DocSim trends based on dataset size change

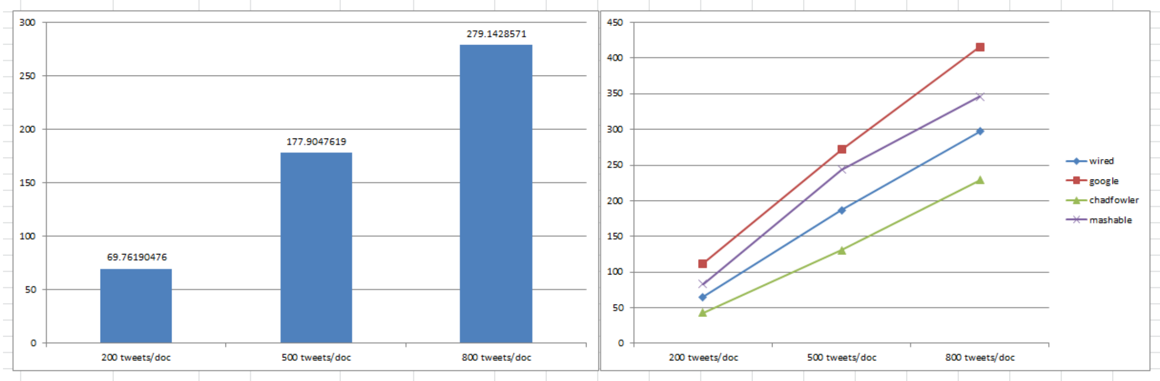


Figure 6.5: WC trends based on dataset size change

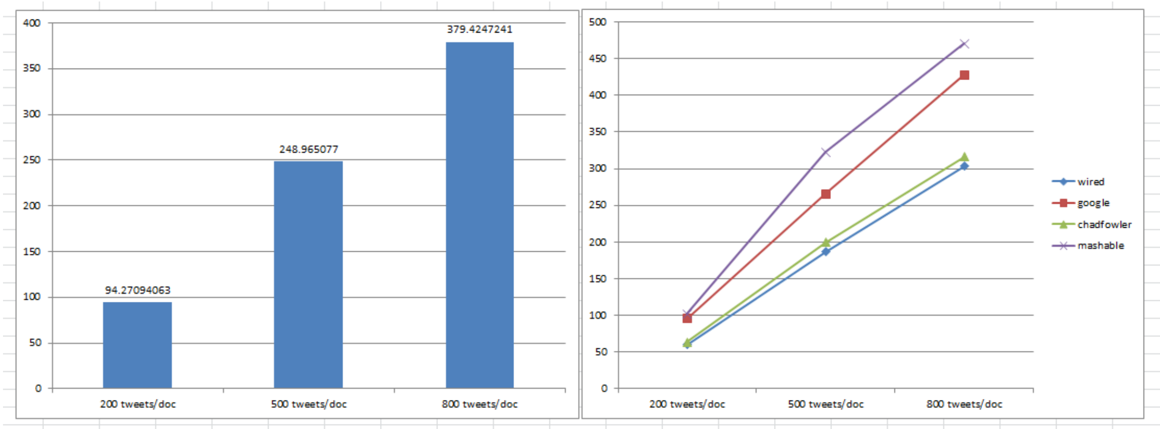


Figure 6.6: XRank trends based on dataset size change

6.3.4 Result D

In this test, 83 users obtain their online influence from XRank algorithm. But Klout Score for several users are not available. Only 69 users have Klout Score. Among those 69 users, 18 are art/design users and 51 are technology users. Table 6.15 represent the technology users rank result. We can see that technology users who have high Klout Score also have high XRank result in technology. This suggests that XRank do have the capability of finding leading influence in social media.

We also notice that the average influence of technology users are higher than that of art/design users. And no art/design users in test dataset has influence higher than 80. This means that Klout Score estimates online influence, ignoring the topic of users belong to. However, XRank is capable of estimate users towards topic. It has already discussed in section 6.3.2.

Table 6.15: XRank result based on full technology and art/design vocabularies, 83 users from both technology and art/design topic. 200 tweets for each document. Column of KloutScore is the Klout Score rank for users. Type column indicate which topic the user belongs to. T represents technology.Users are sorted by Klout Score value from largest to smallest.

ID	ScreenName	Xrank(T)	Xrank(A/D)	KloutScore	Type
27	mashable	314.65635	24.805442	87.76	T
47	TechCrunch	311.95348	18.449761	86.24	T
38	twitter	134.12194	3.9499291	84.87	T
16	google	237.62904	67.77845	82.63	T
28	TheNextWeb	208.51706	17.296444	79.77	T

Table 6.16: XRank result based on full technology and art/design vocabularies, 83 users from both technology and art/design topic. 200 tweets for each document. Column of KloutScore is the Klout Score rank for users. Type column indicate which topic the user belongs to. A represents Art/Design. Users are sorted by Klout Score value from largest to smallest.

ID	ScreenName	Xrank(T)	Xrank(A/D)	KloutScore	Type
68	NewYorker	87.023991	20.200552	78.12	A/D
70	zeldman	22.636614	28.399843	76.83	A/D
37	designmilk	10.396794	216.19601	72.32	A/D
25	LightStalking	7.0199668	63.361454	70.76	A/D
31	designsponge	8.095949	48.006373	70.35	A/D

We also calculate the Spearman correlation coefficient between retweet rate and Klout score. And the coefficient is 0.71. We conjecture that Klout Score algorithm takes retweet rate as an very important parameter and gives retweet rate with a heavy weigh. While for XRank algorithm, we lower the weight of retweet rate. However, retweet rate still plays a very important role in XRank algorithm.

6.4 Summary of Result

As the vocabulary size decreasing, the XRank result will be decreasing as well. As the dataset size decreasing, the XRank result will be decreasing as well. However, the rank order for most user doesn't change much when document consists of more than 500 tweets. This suggests that document contains about 500 tweets is enough for XRank algorithm to estimate online influence. By creating different types of vocabulary, XRank can distinguish users from different topic. In this project, users who have high rank in their own topic will not have higher rank in another different topic. Metadata like retweet rate shows user's capability of enabling other user to have interaction. It's very important aspect of user's influence. XRank rank has low Spearman correlation coefficient with Klout Score(0.4). This means that XRank result and Klout Score result are not similar to each other. XRank algorithm performances better than Klout Score when finding leading influencer in a given topic.

Chapter 7

Conclusion and further work

7.1 Conclusion

In this thesis, we investigated how LSA can be included when estimating online influence in social media. By examining different types of natural language processing methodologies, we found that LSA is capable of discovering latent semantic meaning of document and evaluating document similarities. We also found that word contribution in the reconstructed word matrix also can be utilized to measure the document content. Integrating both document similarity and word contribution, we created a variable called document impact. We associated document impact value with twitter metadata – retweet rate to evaluate online influence. Then we designed and implemented a prototype, to test the new algorithm with different test cases. We named this algorithm XRank. The results have shown that it is possible to include LSA in online influence evaluation.

First, we tested the XRank algorithm on Twitter data with vocabularies of different sizes, to validate how vocabulary size affect XRank result. The result indicates that XRank algorithm depends on vocabulary size and large size vocabulary leads to large XRank result. XRank result nee to be normalize so that they are understandable. Second, through out applying XRank to test data with two types of vocabularies which belong to two different topics. It is proved that XRank algorithm has fairly satisfactory capability of differentiating users that writes about different topics. This capability shows that the XRank algorithm is able to measure and differentiate influence on users by topic. Further more, we changed the document size to verify how document size would affect XRank result. The results show that large document size leads to better rank result. However this leads to a larger computational cost. This test result also suggests that normalization is required to place

XRank result to a given range legitimately. At last, we compare XRank result with Klout Score, the Spearman ρ correlation coefficient shows that the relations between such two rank is low. [That proves that XRank is more like a online influence estimation algorithm in specific topic and Klout score is an overall online influence estimation algorithm.**Rethink**]

7.2 Further work

Document normalization

Kraaij et. al. proved that linguistics stemming commit a significant improvement over linguistic steaming in precision on retrieval performance[22]. Krovetz also noticed that stemming leads to remarkable improvement against non-stemmer in IR system in performance[23]. Despite we removed meaningless items like hyperlinks and stopwords in the documents, there are still more work can be done to clean the documents. One very well known technique is called stemming. Usually, document always contains inflected words like ”thinking” ”thinks” and ”thought”. These inflected words can be stemmed by stemming algorithms. Stemming algorithm stems inflected words into the root form. Stemming enables us to concentrate on the meaning of the words and save computational resources.

Another aspect that can improve the topic classification in XRank is to spell check the documents. It’s unavoidable that there are words that are spelled wrong in a large document base. Misspelled words introduces corruption to the IR systems when retrieving informations from documents[31]. In this project, in order to remove the wrong words, we filtered all words that only words appears more than 2 times in all documents are collected and analysed. However, this filtering step can not be filter all wrong words. as this reason, we need to check all words in document and make sure that the words analysed are correct.

Improve term-document matrix

When we constructed the term-document matrix, we simply counted the word occurrence and set them into the matrix. Because of reason, we have to apply our tests to documents with the same number of tweets. Event though, we can not guarantee that all documents are with same length. Short documents have less words than long documents have, hence short document will have disadvantages. To deduce such weakness, TF-IDF can be introduced to LSA. That suggests that when constructing term-document matrix, instead of creating matrix cell value with word occurrence, we use TF-IDF value to fill cells in term-document matrix. TF-IDF can find relevant and valuable keywords from document and LSA to find latent semantic relations[9].

Improve third-party SVD library to support large dimension sparse matrix

In this project, we used the python svd library called Scipy and Numpy. The problem for Scipy SVD is that it doesn't support decomposition on matrix with a dimension lager than 13000. The solution is either we resolve the problem by cleaning documents, or we can find another SVD library that can handle large sparse matrix decomposition.

Improve document similarity approximation rank

LSA utilizes rank approximation to get document similarity. In this project, we choose k=2 to calculate document similarity.This value applies the disparity among all documents. But we can not guarantee that 2 is the best value. So further work is to find the best or empirically best value for rank approximation.

Add more proper parameters to XRank

As we mentioned in previous paragraph, we only use two parameters in XRank algorithm. On social media, there are more parameters can be used like active followers and follower's influence, etc. We believe that more valuable parameters are added to XRank algorithm, better rank result would be obtained.

Normalize XRank result

In this thesis we have seen that XRank result are sometimes very large to thousand and small to about 1. In order to make the result more understandable, a simple normalization process can be applied.

Appendix A

Acronyms

LSA Latent Semantic Analysis

LSI Latent Semantic Indexing

API Application Programming Interface

NLP Natural Language Processing

TT Turing Test

IG Imitation Game

VSM Vector Space Model

TF-IDF Term Frequency − Inverse Document Frequency

IR Information Retrieval

SVD Singular Value Decomposition

WC Word Contribution

OI Online Influence

RC Retweet Count

TC Tweet Count

CS Cosine Similarity

DocSim Document similarity

PWC Positive Word Contribution

DI Document Impact

RTR Retweet Rate

URL Uniform Resource Locator

UI User Interface

Appendix B

SVD mathematical example

This is the classical SVD exmaple from Deerwsters's paper[12]. This technical example contains titles of nine documents. Only word occurrences no less than twice will be indexed in the term-document matrix. As premise knowledge, five titles about human-computer interaction (marked as c1-c5) and four titles about graph theory (marked as m1-m4). The cell values in term-document matrix are simply occurrence times in titles.

Titles

c1: Human machine interface for Lab ABC computer applications
c2: A survey of user opinion of computer system response time
c3: The EPS user interface management system
c4: System and human system engineering testing of EPS
c5: Relation of user-perceived response time to error measurement

m1: The generation of random, binary, unordered trees
m2: The intersection graph of paths in trees
m3: Graph minors IV: Widths of trees and well-quasi-ordering
m4: Graph minors:A survey

And term-document matrix would be like:

$$\begin{array}{cccccccccc}
 & c1 & c2 & c3 & c4 & c5 & m1 & m2 & m3 & m4 \\
human & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
interface & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
computer & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
user & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
system & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
response & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
time & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
EPS & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
survey & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
trees & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
graph & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
minors & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}$$

Denoted this term-document matrix as M, and A can be decomposed by Singular Value Decomposition into three matrices.

$$A = U_0 \Sigma_0 V_0^T$$

U_0 is nine dimensional left-singular vectors for 12 terms, S_0 is the diagonal matrix of nine singular values with decreased order, and D_0 is the nine dimensional right singular vector for nine document.

$$U_0 =$$

$$\begin{matrix}
0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\
0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\
0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\
0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\
0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\
0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\
0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\
0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\
0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\
0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\
0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\
0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18
\end{matrix}$$

$$\Sigma_0 =$$

$$\begin{matrix}
3.34 & & & & & & & & \\
& 2.54 & & & & & & & \\
& & 2.35 & & & & & & \\
& & & 1.64 & & & & & \\
& & & & 1.50 & & & & \\
& & & & & 1.31 & & & \\
& & & & & & 0.85 & & \\
& & & & & & & 0.56 & \\
& & & & & & & & 0.36
\end{matrix}$$

$$V_0 =$$

$$\begin{matrix}
0.20 & -0.06 & 0.11 & -0.95 & 0.05 & -0.08 & 0.18 & -0.01 & -0.06 \\
0.61 & 0.17 & -0.50 & -0.03 & -0.21 & -0.26 & -0.43 & 0.05 & 0.24 \\
0.46 & -0.13 & 0.21 & 0.04 & 0.38 & 0.72 & -0.24 & 0.01 & 0.02 \\
0.54 & -0.23 & 0.57 & 0.27 & -0.21 & -0.37 & 0.26 & -0.02 & -0.08 \\
0.28 & 0.11 & -0.51 & 0.15 & 0.33 & 0.03 & 0.67 & -0.06 & -0.26 \\
0.00 & 0.19 & 0.10 & 0.02 & 0.39 & -0.30 & -0.34 & 0.45 & -0.62 \\
0.01 & 0.44 & 0.19 & 0.02 & 0.35 & -0.21 & -0.15 & -0.76 & 0.02 \\
0.02 & 0.62 & 0.25 & 0.01 & 0.15 & 0.00 & 0.25 & 0.45 & 0.52 \\
0.08 & 0.53 & 0.08 & -0.03 & -0.60 & 0.36 & 0.04 & -0.07 & -0.45
\end{matrix}$$

Depend on the operational criteria, the value of dimensions deduction k is set as 2. Basically, a large k can cover every details of the data structure, while under the need of this retrieval example, small k value can eliminate sampling errors and ignore unimportant details.

$$A \approx \hat{A} = U \Sigma V^T$$

$A =$

$$\begin{array}{ccccccccccccc}
 & T & & \Sigma & & & & & V^T & & & & \\
0.22 & -0.11 & 3.34 & & 0.20 & 0.61 & 0.64 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\
0.20 & -0.07 & & 2,54 & -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \\
0.24 & 0.04 & & & & & & & & & & & \\
0.40 & 0.06 & & & & & & & & & & & \\
0.64 & -0.17 & & & & & & & & & & & \\
0.27 & 0.11 & & & & & & & & & & & \\
0.27 & 0.11 & & & & & & & & & & & \\
0.30 & -0.14 & & & & & & & & & & & \\
0.21 & 0.27 & & & & & & & & & & & \\
0.01 & 0.49 & & & & & & & & & & & \\
0.04 & 0.62 & & & & & & & & & & & \\
0.03 & 0.45 & & & & & & & & & & &
\end{array}$$

The product of these three matrix will produce $\hat{A}$

$$\hat{A} =$$

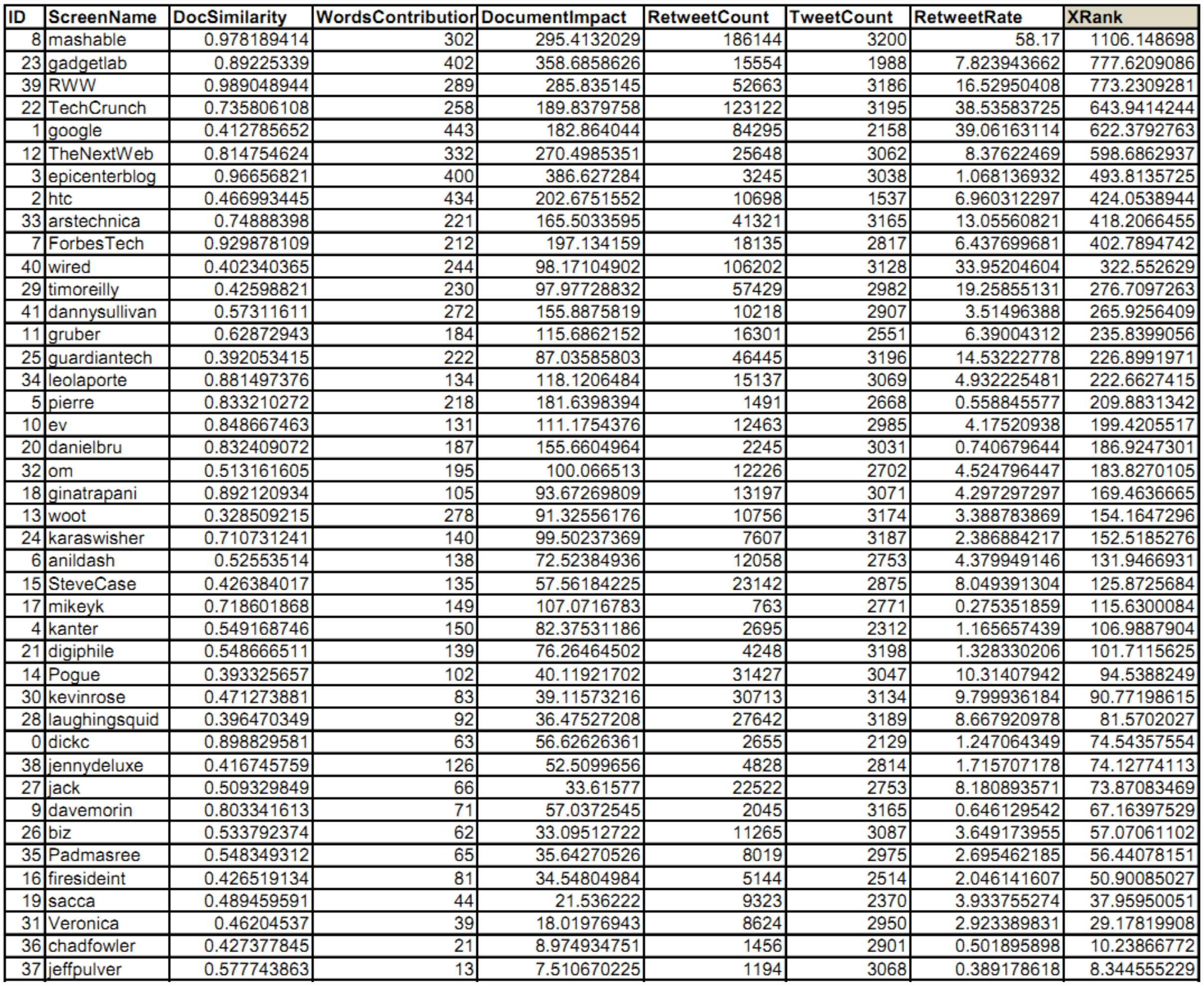
$$\begin{array}{cccccccccc}
human & 0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\
interface & 0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\
computer & 0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\
user & 0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\
sysyem & 0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\
response & 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\
time & 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\
EPS & 0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\
sruvey & 0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\
trees & -0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\
graph & -0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\
minors & -0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62
\end{array}$$

It has to mentioned that this value in this matrix is not exactly match the terms in document, and the value would be getting close and close as more and more singular value are kept.

Appendix C

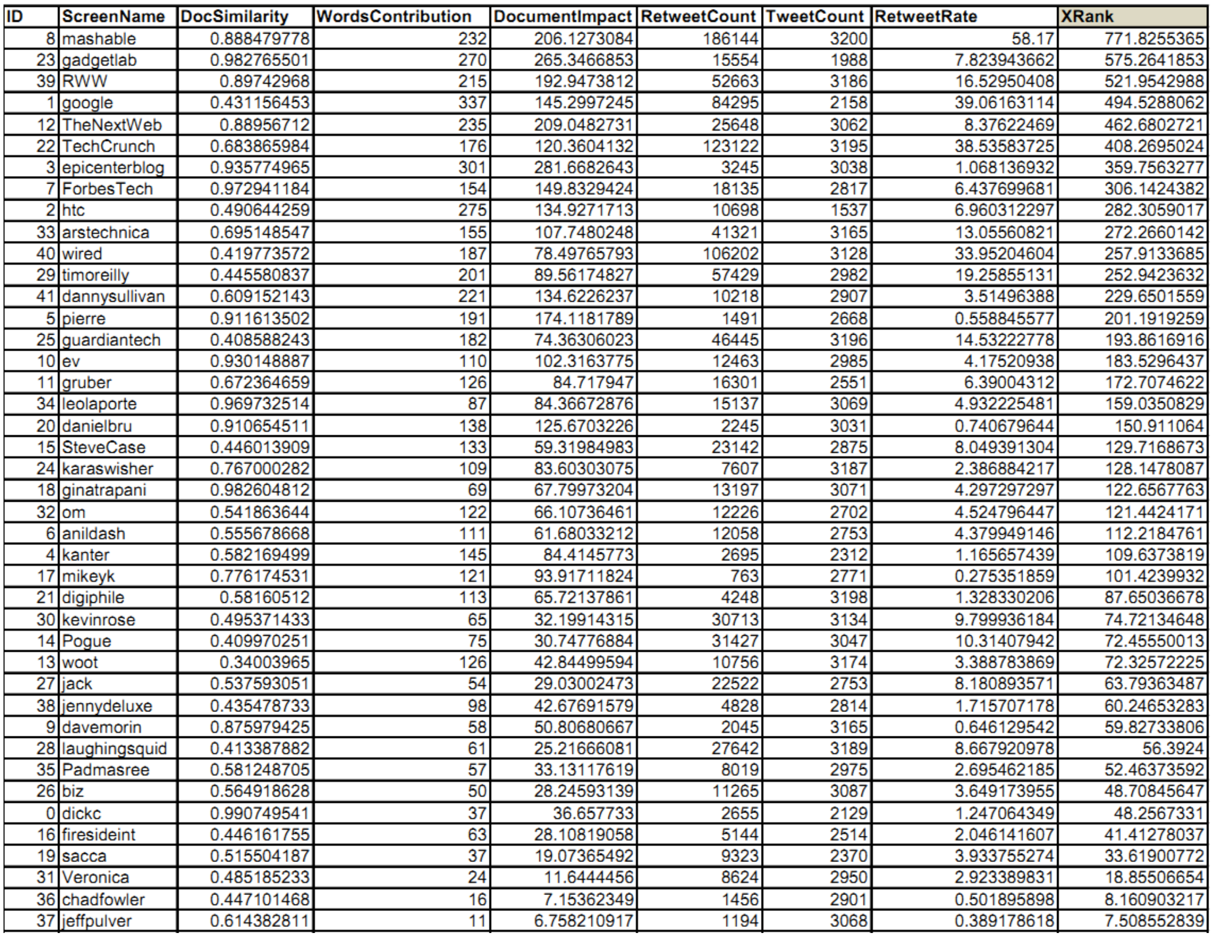
XRank algorithm test result based on variance of vocabulary size

This appendix show the XRank algorithm test result based on variance of technology vocabulary .



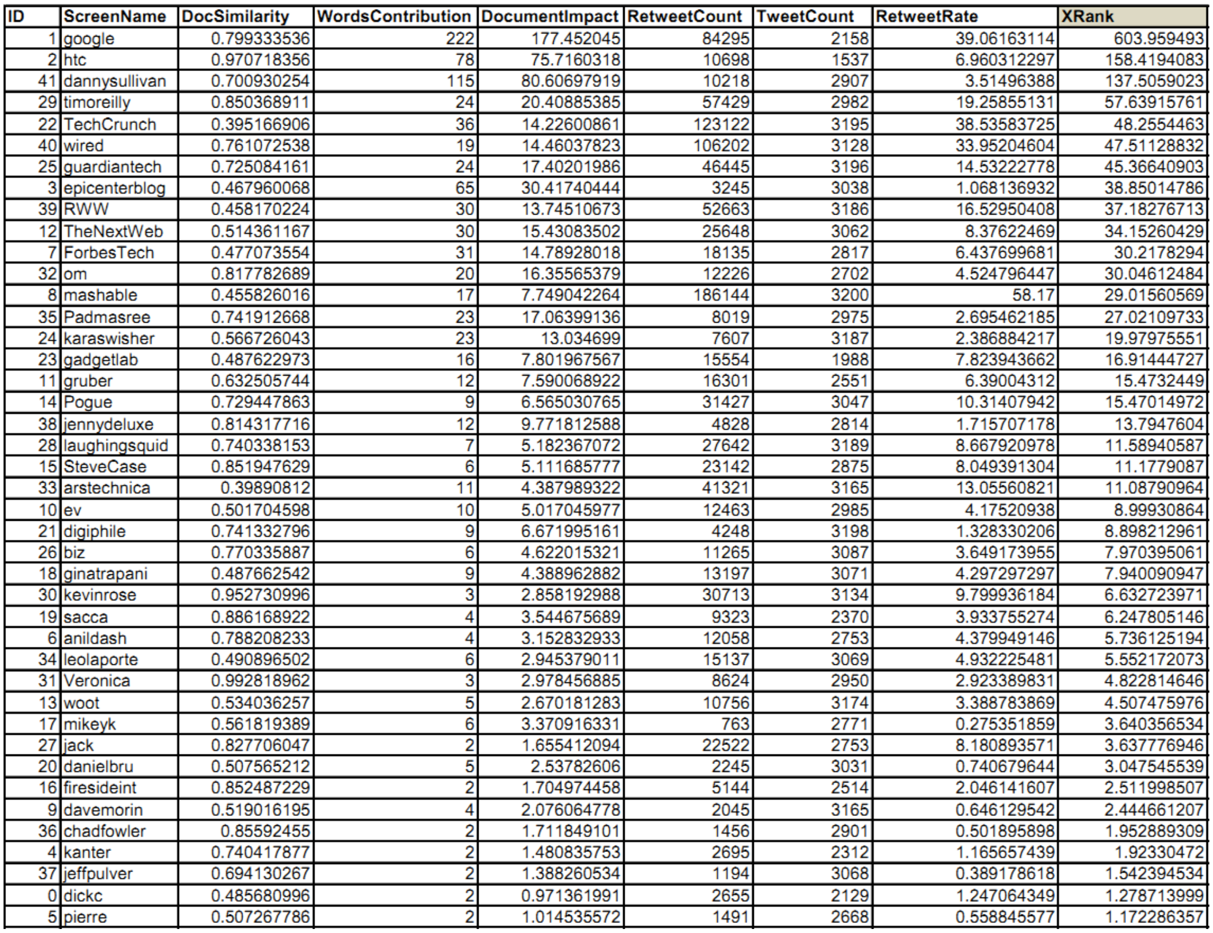
ID	ScreenName	DocSimilarity	WordsContributior	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
8	mashable	0.978189414	302	295.4132029	186144	3200	58.17	1106.148698
23	gadgetlab	0.89225339	402	358.6858626	15554	1988	7.823943662	777.6209086
39	RWW	0.989048944	289	285.835145	52663	3186	16.52950408	773.2309281
22	TechCrunch	0.735806108	258	189.8379758	123122	3195	38.53583725	643.9414244
1	google	0.412785652	443	182.864044	84295	2158	39.06163114	622.3792763
12	TheNextWeb	0.814754624	332	270.4985351	25648	3062	8.37622469	598.6862937
3	epicenterblog	0.96656821	400	386.627284	3245	3038	1.068136932	493.8135725
2	htc	0.466993445	434	202.6751552	10698	1537	6.960312297	424.0538944
33	arstechnica	0.74888398	221	165.5033595	41321	3165	13.05560821	418.2066455
7	ForbesTech	0.929878109	212	197.134159	18135	2817	6.437699681	402.7894742
40	wired	0.402340365	244	98.17104902	106202	3128	33.95204604	322.552629
29	timoreilly	0.42598821	230	97.97728832	57429	2982	19.25855131	276.7097263
41	dannysullivan	0.57311611	272	155.8875819	10218	2907	3.51496388	265.9256409
11	gruber	0.62872943	184	115.6862152	16301	2551	6.39004312	235.8399056
25	guardiantech	0.392053415	222	87.03585803	46445	3196	14.53222778	226.8991971
34	leolaporte	0.881497376	134	118.1206484	15137	3069	4.932225481	222.6627415
5	pierre	0.833210272	218	181.6398394	1491	2668	0.558845577	209.8831342
10	ev	0.848667463	131	111.1754376	12463	2985	4.17520938	199.4205517
20	danielbru	0.832409072	187	155.6604964	2245	3031	0.740679644	186.9247301
32	om	0.513161605	195	100.066513	12226	2702	4.524796447	183.8270105
18	ginatrapani	0.892120934	105	93.67269809	13197	3071	4.297297297	169.4636665
13	woot	0.328509215	278	91.32556176	10756	3174	3.388783869	154.1647296
24	karaswisher	0.710731241	140	99.50237369	7607	3187	2.386884217	152.5185276
6	anildash	0.52553514	138	72.52384936	12058	2753	4.379949146	131.9466931
15	SteveCase	0.426384017	135	57.56184225	23142	2875	8.049391304	125.8725684
17	mikeyk	0.718601868	149	107.0716783	763	2771	0.275351859	115.6300084
4	kanter	0.549168746	150	82.37531186	2695	2312	1.165657439	106.9887904
21	digiphile	0.548666511	139	76.26464502	4248	3198	1.328330206	101.7115625
14	Pogue	0.393325657	102	40.11921702	31427	3047	10.31407942	94.5388249
30	kevinrose	0.471273881	83	39.11573216	30713	3134	9.799936184	90.77198615
28	laughingsquid	0.396470349	92	36.47527208	27642	3189	8.667920978	81.5702027
0	dickc	0.898829581	63	56.62626361	2655	2129	1.247064349	74.54357554
38	jennydeluxe	0.416745759	126	52.5099656	4828	2814	1.715707178	74.12774113
27	jack	0.509329849	66	33.61577	22522	2753	8.180893571	73.87083469
9	davemorin	0.803341613	71	57.0372545	2045	3165	0.646129542	67.16397529
26	biz	0.533792374	62	33.09512722	11265	3087	3.649173955	57.07061102
35	Padmasree	0.548349312	65	35.64270526	8019	2975	2.695462185	56.44078151
16	firesideint	0.426519134	81	34.54804984	5144	2514	2.046141607	50.90085027
19	sacca	0.489459591	44	21.536222	9323	2370	3.933755274	37.95950051
31	Veronica	0.46204537	39	18.01976943	8624	2950	2.923389831	29.17819908
36	chadfowler	0.427377845	21	8.974934751	1456	2901	0.501895898	10.23866772
37	jeffpulver	0.577743863	13	7.510670225	1194	3068	0.389178618	8.344555229

Figure C.1: XRank result with full size of technology vocabulary, 42 users from technology topic, 500 tweets for each user.



ID	ScreenName	DocSimilarity	WordsContribution	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
8	mashable	0.888479778	232	206.1273084	186144	3200	58.17	771.8255365
23	gadgetlab	0.982765501	270	265.3466853	15554	1988	7.823943662	575.2641853
39	RWW	0.89742968	215	192.9473812	52663	3186	16.52950408	521.9542988
1	google	0.431156453	337	145.2997245	84295	2158	39.06163114	494.5288062
12	TheNextWeb	0.88956712	235	209.0482731	25648	3062	8.37622469	462.6802721
22	TechCrunch	0.683865984	176	120.3604132	123122	3195	38.53583725	408.2695024
3	epicenterblog	0.935774965	301	281.6682643	3245	3038	1.068136932	359.7563277
7	ForbesTech	0.972941184	154	149.8329424	18135	2817	6.437699681	306.1424382
2	htc	0.490644259	275	134.9271713	10698	1537	6.960312297	282.3059017
33	arstechnica	0.695148547	155	107.7480248	41321	3165	13.05560821	272.2660142
40	wired	0.419773572	187	78.49765793	106202	3128	33.95204604	257.9133685
29	timoreilly	0.445580837	201	89.56174827	57429	2982	19.25855131	252.9423632
41	dannysullivan	0.609152143	221	134.6226237	10218	2907	3.51496388	229.6501559
5	pierre	0.911613502	191	174.1181789	1491	2668	0.558845577	201.1919259
25	guardiantech	0.408588243	182	74.36306023	46445	3196	14.53222778	193.8616916
10	ev	0.930148887	110	102.3163775	12463	2985	4.17520938	183.5296437
11	gruber	0.672364659	126	84.717947	16301	2551	6.39004312	172.7074622
34	leolaporte	0.969732514	87	84.36672876	15137	3069	4.932225481	159.0350829
20	danielbru	0.910654511	138	125.6703226	2245	3031	0.740679644	150.911064
15	SteveCase	0.446013909	133	59.31984983	23142	2875	8.049391304	129.7168673
24	karaswisher	0.767000282	109	83.60303075	7607	3187	2.386884217	128.1478087
18	ginatrapani	0.982604812	69	67.79973204	13197	3071	4.297297297	122.6567763
32	om	0.541863644	122	66.10736461	12226	2702	4.524796447	121.4424171
6	anildash	0.555678668	111	61.68033212	12058	2753	4.379949146	112.2184761
4	kanter	0.582169499	145	84.4145773	2695	2312	1.165657439	109.6373819
17	mikeyk	0.776174531	121	93.91711824	763	2771	0.275351859	101.4239932
21	digiphile	0.58160512	113	65.72137861	4248	3198	1.328330206	87.65036678
30	kevinrose	0.495371433	65	32.19914315	30713	3134	9.799936184	74.72134648
14	Pogue	0.409970251	75	30.74776884	31427	3047	10.31407942	72.45550013
13	woot	0.34003965	126	42.84499594	10756	3174	3.388783869	72.32572225
27	jack	0.537593051	54	29.03002473	22522	2753	8.180893571	63.79363487
38	jennydeluxe	0.435478733	98	42.67691579	4828	2814	1.715707178	60.24653283
9	davemorin	0.875979425	58	50.80680667	2045	3165	0.646129542	59.82733806
28	laughingsquid	0.413387882	61	25.21666081	27642	3189	8.667920978	56.3924
35	Padmasree	0.581248705	57	33.13117619	8019	2975	2.695462185	52.46373592
26	biz	0.564918628	50	28.24593139	11265	3087	3.649173955	48.70845647
0	dickc	0.990749541	37	36.657733	2655	2129	1.247064349	48.2567331
16	firesideint	0.446161755	63	28.10819058	5144	2514	2.046141607	41.41278037
19	sacca	0.515504187	37	19.07365492	9323	2370	3.933755274	33.61900772
31	Veronica	0.485185233	24	11.6444456	8624	2950	2.923389831	18.85506654
36	chadfowler	0.447101468	16	7.15362349	1456	2901	0.501895898	8.160903217
37	jeffpulver	0.614382811	11	6.758210917	1194	3068	0.389178618	7.508552839

Figure C.2: XRank result with half size of technology vocabulary, 42 users from technology topic, 500 tweets for each user

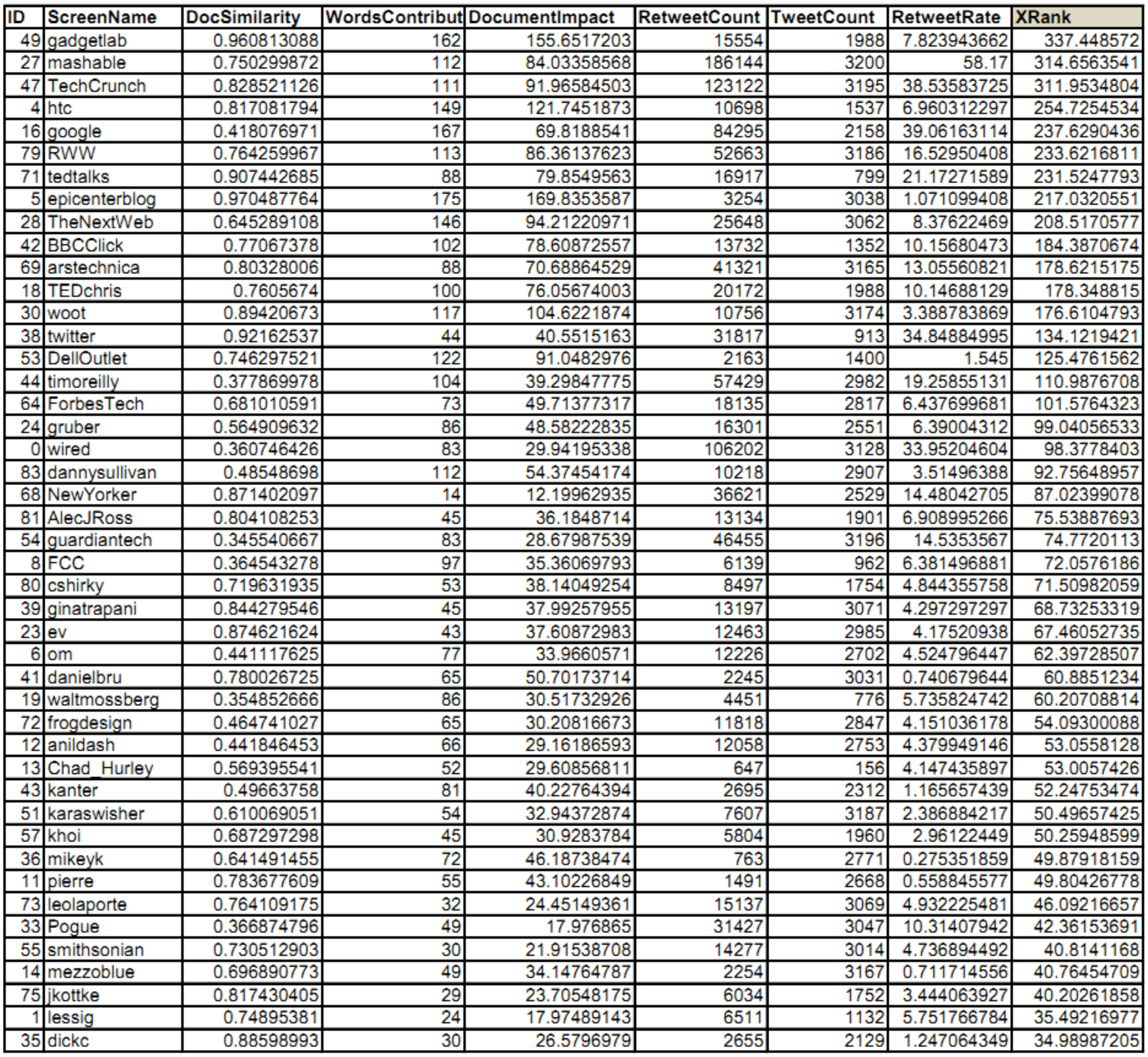


ID	ScreenName	DocSimilarity	WordsContribution	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
1	google	0.799333536	222	177.452045	84295	2158	39.06163114	603.959493
2	htc	0.970718356	78	75.7160318	10698	1537	6.960312297	158.4194083
41	dannysullivan	0.700930254	115	80.60697919	10218	2907	3.51496388	137.5059023
29	timoreilly	0.850368911	24	20.40885385	57429	2982	19.25855131	57.63915761
22	TechCrunch	0.395166906	36	14.22600861	123122	3195	38.53583725	48.2554463
40	wired	0.761072538	19	14.46037823	106202	3128	33.95204604	47.51128832
25	guardiantech	0.725084161	24	17.40201986	46445	3196	14.53222778	45.36640903
3	epicenterblog	0.467960068	65	30.41740444	3245	3038	1.068136932	38.85014786
39	RWW	0.458170224	30	13.74510673	52663	3186	16.52950408	37.18276713
12	TheNextWeb	0.514361167	30	15.43083502	25648	3062	8.37622469	34.15260429
7	ForbesTech	0.477073554	31	14.78928018	18135	2817	6.437699681	30.2178294
32	om	0.817782689	20	16.35565379	12226	2702	4.524796447	30.04612484
8	mashable	0.455826016	17	7.749042264	186144	3200	58.17	29.01560569
35	Padmasree	0.741912668	23	17.06399136	8019	2975	2.695462185	27.02109733
24	karaswisher	0.566726043	23	13.034699	7607	3187	2.386884217	19.97975551
23	gadgetlab	0.487622973	16	7.801967567	15554	1988	7.823943662	16.91444727
11	gruber	0.632505744	12	7.590068922	16301	2551	6.39004312	15.4732449
14	Pogue	0.729447863	9	6.565030765	31427	3047	10.31407942	15.47014972
38	jennydeluxe	0.814317716	12	9.771812588	4828	2814	1.715707178	13.7947604
28	laughingsquid	0.740338153	7	5.182367072	27642	3189	8.667920978	11.58940587
15	SteveCase	0.851947629	6	5.111685777	23142	2875	8.049391304	11.1779087
33	arstechnica	0.39890812	11	4.387989322	41321	3165	13.05560821	11.08790964
10	ev	0.501704598	10	5.017045977	12463	2985	4.17520938	8.99930864
21	digiphile	0.741332796	9	6.671995161	4248	3198	1.328330206	8.898212961
26	biz	0.770335887	6	4.622015321	11265	3087	3.649173955	7.970395061
18	ginatrapani	0.487662542	9	4.388962882	13197	3071	4.297297297	7.940090947
30	kevinrose	0.952730996	3	2.858192988	30713	3134	9.799936184	6.632723971
19	sacca	0.886168922	4	3.544675689	9323	2370	3.933755274	6.247805146
6	anildash	0.788208233	4	3.152832933	12058	2753	4.379949146	5.736125194
34	leolaporte	0.490896502	6	2.945379011	15137	3069	4.932225481	5.552172073
31	Veronica	0.992818962	3	2.978456885	8624	2950	2.923389831	4.822814646
13	woot	0.534036257	5	2.670181283	10756	3174	3.388783869	4.507475976
17	mikeyk	0.561819389	6	3.370916331	763	2771	0.275351859	3.640356534
27	jack	0.827706047	2	1.655412094	22522	2753	8.180893571	3.637776946
20	danielbru	0.507565212	5	2.53782606	2245	3031	0.740679644	3.047545539
16	firesideint	0.852487229	2	1.704974458	5144	2514	2.046141607	2.511998507
9	davemorin	0.519016195	4	2.076064778	2045	3165	0.646129542	2.444661207
36	chadfowler	0.85592455	2	1.711849101	1456	2901	0.501895898	1.952889309
4	kanter	0.740417877	2	1.480835753	2695	2312	1.165657439	1.92330472
37	jeffpulver	0.694130267	2	1.388260534	1194	3068	0.389178618	1.542394534
0	dickc	0.485680996	2	0.971361991	2655	2129	1.247064349	1.278713999
5	pierre	0.507267786	2	1.014535572	1491	2668	0.558845577	1.172286357

Figure C.3: XRank result with technology vocabulary consists of 9 words, 42 users from technology topic, 500 tweets for each user

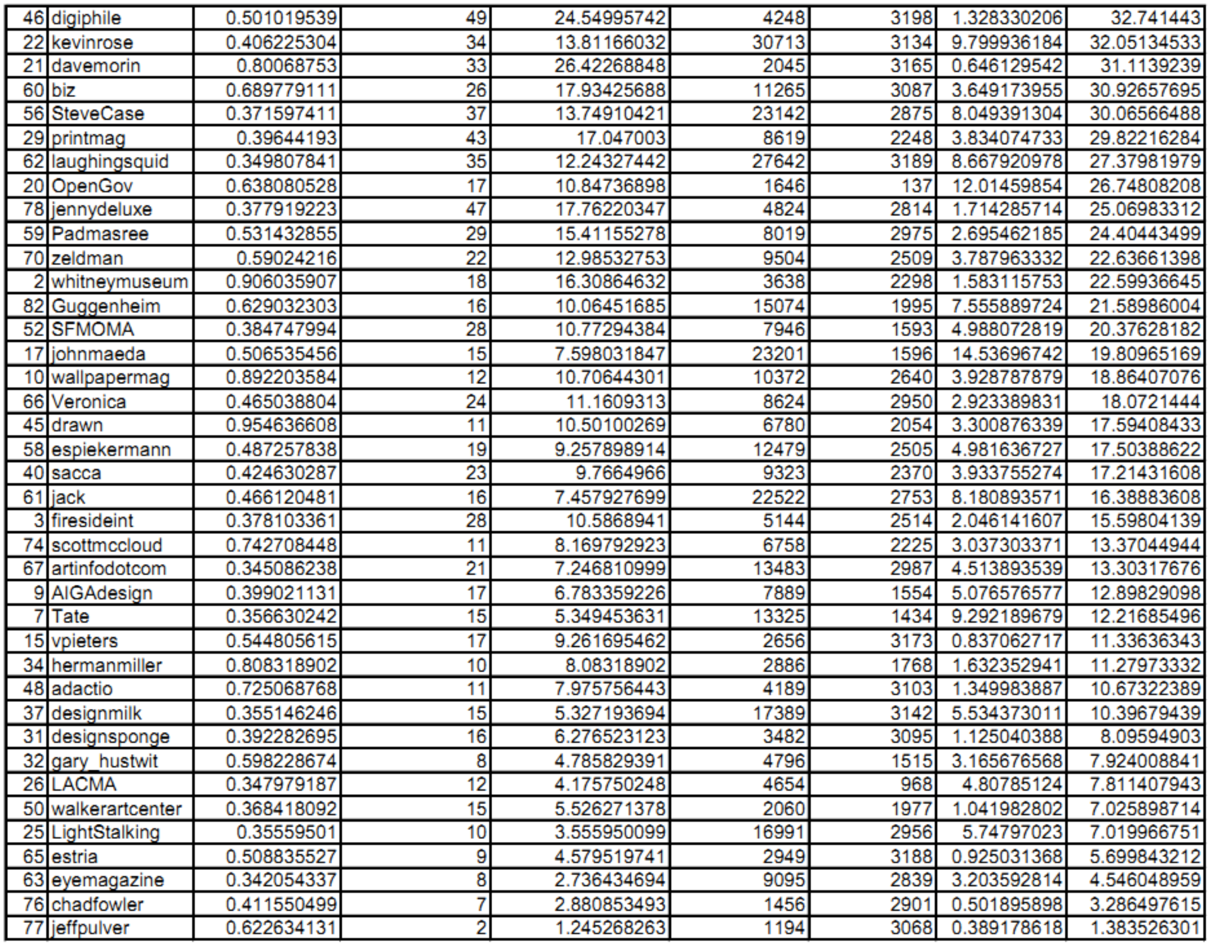
Appendix D

XRank algorithm test result based on variance of vocabulary topic



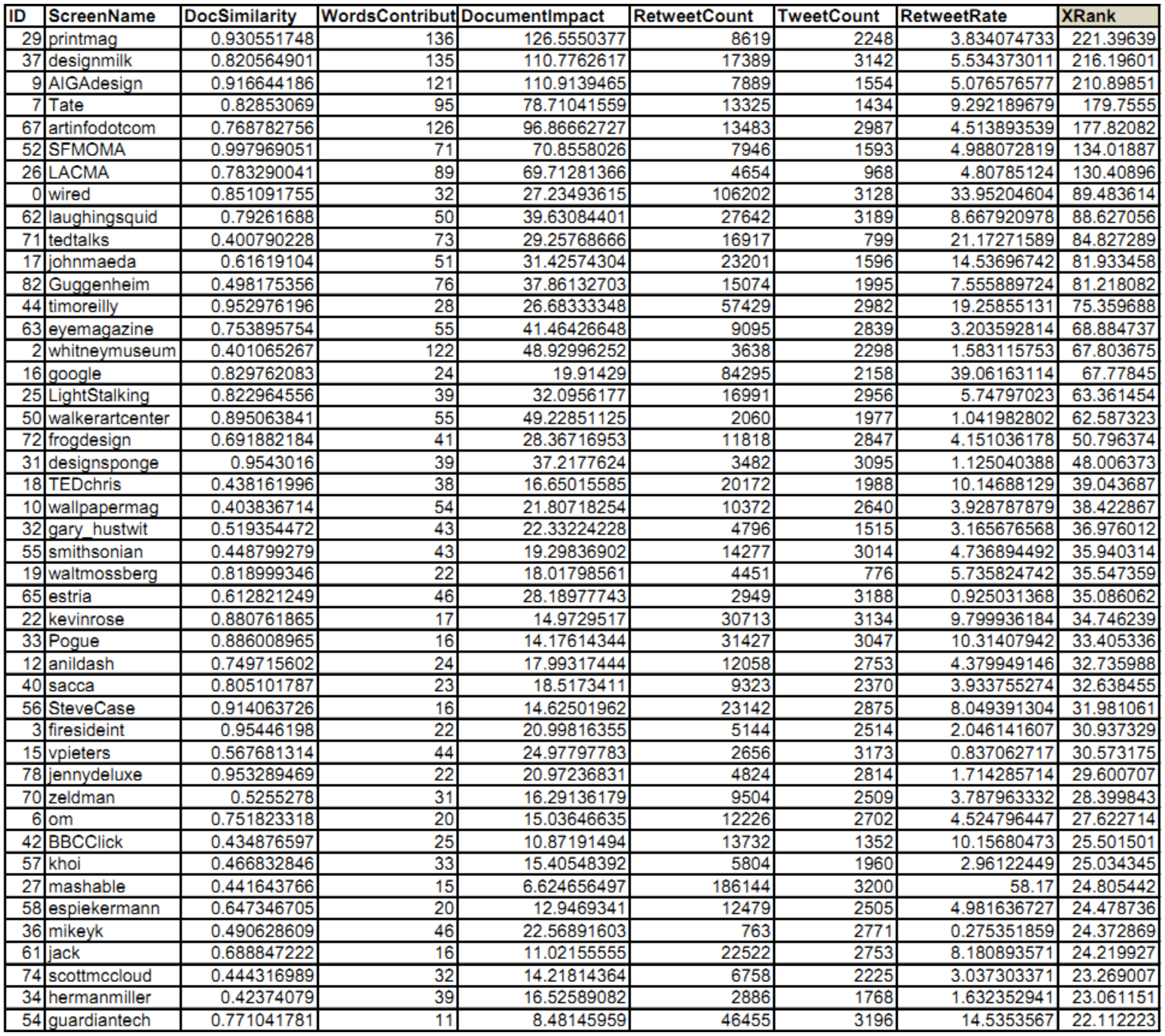
ID	ScreenName	DocSimilarity	WordsContribut	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
49	gadgetlab	0.960813088	162	155.6517203	15554	1988	7.823943662	337.448572
27	mashable	0.750299872	112	84.03358568	186144	3200	58.17	314.6563541
47	TechCrunch	0.828521126	111	91.96584503	123122	3195	38.53583725	311.9534804
4	htc	0.817081794	149	121.7451873	10698	1537	6.960312297	254.7254534
16	google	0.418076971	167	69.8188541	84295	2158	39.06163114	237.6290436
79	RWW	0.764259967	113	86.36137623	52663	3186	16.52950408	233.6216811
71	tedtalks	0.907442685	88	79.8549563	16917	799	21.17271589	231.5247793
5	epicenterblog	0.970487764	175	169.8353587	3254	3038	1.071099408	217.0320551
28	TheNextWeb	0.645289108	146	94.21220971	25648	3062	8.37622469	208.5170577
42	BBCClick	0.77067378	102	78.60872557	13732	1352	10.15680473	184.3870674
69	arstechnica	0.80328006	88	70.68864529	41321	3165	13.05560821	178.6215175
18	TEDchris	0.7605674	100	76.05674003	20172	1988	10.14688129	178.348815
30	woot	0.89420673	117	104.6221874	10756	3174	3.388783869	176.6104793
38	twitter	0.92162537	44	40.5515163	31817	913	34.84884995	134.1219421
53	DellOutlet	0.746297521	122	91.0482976	2163	1400	1.545	125.4761562
44	timoreilly	0.377869978	104	39.29847775	57429	2982	19.25855131	110.9876708
64	ForbesTech	0.681010591	73	49.71377317	18135	2817	6.437699681	101.5764323
24	gruber	0.564909632	86	48.58222835	16301	2551	6.39004312	99.04056533
0	wired	0.360746426	83	29.94195338	106202	3128	33.95204604	98.3778403
83	dannysullivan	0.48548698	112	54.37454174	10218	2907	3.51496388	92.75648957
68	NewYorker	0.871402097	14	12.19962935	36621	2529	14.48042705	87.02399078
81	AlecJRoss	0.804108253	45	36.1848714	13134	1901	6.908995266	75.53887693
54	guardiantech	0.345540667	83	28.67987539	46455	3196	14.5353567	74.7720113
8	FCC	0.364543278	97	35.36069793	6139	962	6.381496881	72.0576186
80	cshirky	0.719631935	53	38.14049254	8497	1754	4.844355758	71.50982059
39	ginatrapani	0.844279546	45	37.99257955	13197	3071	4.297297297	68.73253319
23	ev	0.874621624	43	37.60872983	12463	2985	4.17520938	67.46052735
6	om	0.441117625	77	33.9660571	12226	2702	4.524796447	62.39728507
41	danielbru	0.780026725	65	50.70173714	2245	3031	0.740679644	60.8851234
19	waltmossberg	0.354852666	86	30.51732926	4451	776	5.735824742	60.20708814
72	frogdesign	0.464741027	65	30.20816673	11818	2847	4.151036178	54.09300088
12	anildash	0.441846453	66	29.16186593	12058	2753	4.379949146	53.0558128
13	Chad_Hurley	0.569395541	52	29.60856811	647	156	4.147435897	53.0057426
43	kanter	0.49663758	81	40.22764394	2695	2312	1.165657439	52.24753474
51	karaswisher	0.610069051	54	32.94372874	7607	3187	2.386884217	50.49657425
57	khoi	0.687297298	45	30.9283784	5804	1960	2.96122449	50.25948599
36	mikeyk	0.641491455	72	46.18738474	763	2771	0.275351859	49.87918159
11	pierre	0.783677609	55	43.10226849	1491	2668	0.558845577	49.80426778
73	leolaporte	0.764109175	32	24.45149361	15137	3069	4.932225481	46.09216657
33	Pogue	0.366874796	49	17.976865	31427	3047	10.31407942	42.36153691
55	smithsonian	0.730512903	30	21.91538708	14277	3014	4.736894492	40.8141168
14	mezzoblue	0.696890773	49	34.14764787	2254	3167	0.711714556	40.76454709
75	jkottke	0.817430405	29	23.70548175	6034	1752	3.444063927	40.20261858
1	lessig	0.74895381	24	17.97489143	6511	1132	5.751766784	35.49216977
35	dickc	0.88598993	30	26.5796979	2655	2129	1.247064349	34.98987205

Figure D.1: XRank result with 83 users from both art/design and technology topic, 200 tweets for each user, full technology vocabulary. Part 1



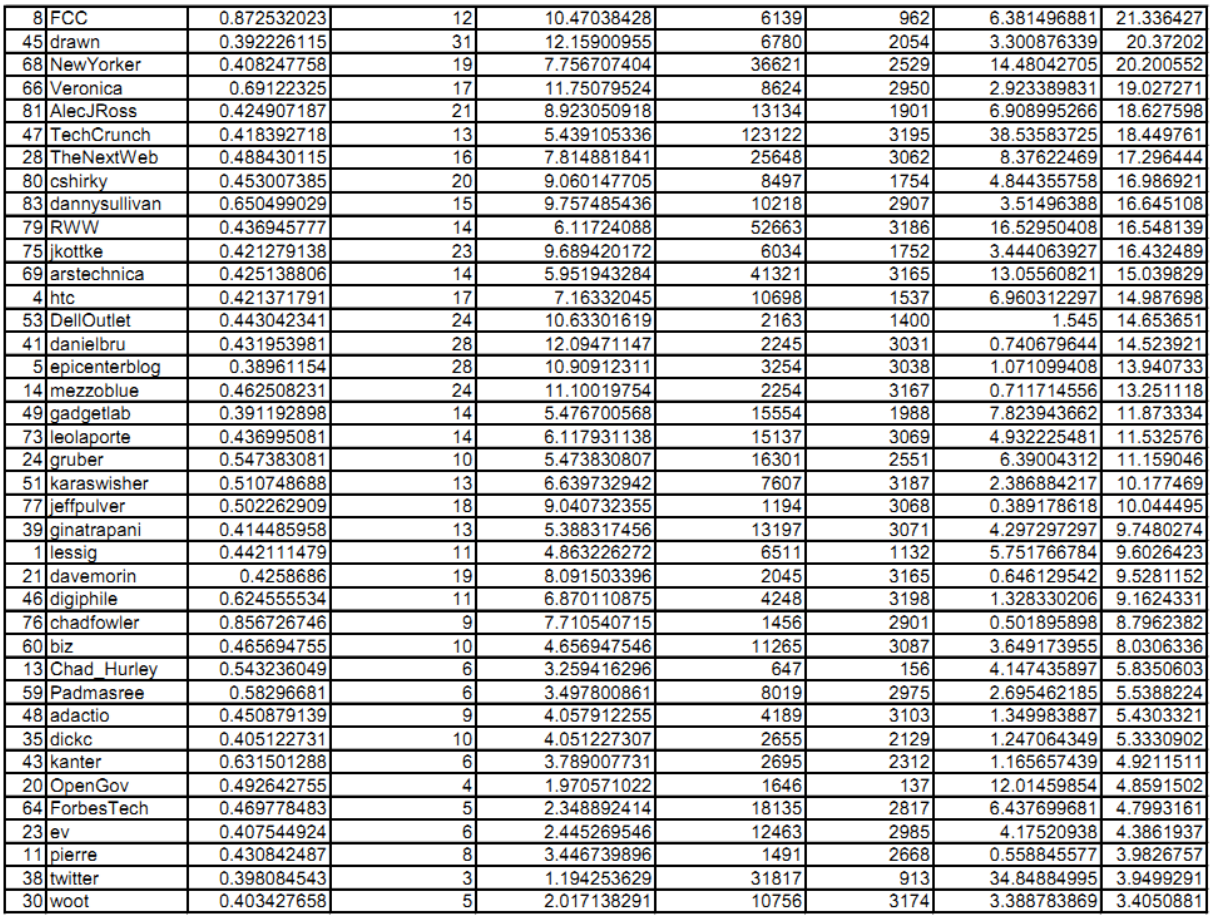
46	digiphile	0.501019539	49	24.54995742	4248	3198	1.328330206	32.741443
22	kevinrose	0.406225304	34	13.81166032	30713	3134	9.799936184	32.05134533
21	davemorin	0.80068753	33	26.42268848	2045	3165	0.646129542	31.1139239
60	biz	0.689779111	26	17.93425688	11265	3087	3.649173955	30.92657695
56	SteveCase	0.371597411	37	13.74910421	23142	2875	8.049391304	30.06566488
29	printmag	0.39644193	43	17.047003	8619	2248	3.834074733	29.82216284
62	laughingsquid	0.349807841	35	12.24327442	27642	3189	8.667920978	27.37981979
20	OpenGov	0.638080528	17	10.84736898	1646	137	12.01459854	26.74808208
78	jennydeluxe	0.377919223	47	17.76220347	4824	2814	1.714285714	25.06983312
59	Padmasree	0.531432855	29	15.41155278	8019	2975	2.695462185	24.40443499
70	zeldman	0.59024216	22	12.98532753	9504	2509	3.787963332	22.63661398
2	whitneymuseum	0.906035907	18	16.30864632	3638	2298	1.583115753	22.59936645
82	Guggenheim	0.629032303	16	10.06451685	15074	1995	7.555889724	21.58986004
52	SFMOMA	0.384747994	28	10.77294384	7946	1593	4.988072819	20.37628182
17	johnmaeda	0.506535456	15	7.598031847	23201	1596	14.53696742	19.80965169
10	wallpapermag	0.892203584	12	10.70644301	10372	2640	3.928787879	18.86407076
66	Veronica	0.465038804	24	11.1609313	8624	2950	2.923389831	18.0721444
45	drawn	0.954636608	11	10.50100269	6780	2054	3.300876339	17.59408433
58	espiekermann	0.487257838	19	9.257898914	12479	2505	4.981636727	17.50388622
40	sacca	0.424630287	23	9.7664966	9323	2370	3.933755274	17.21431608
61	jack	0.466120481	16	7.457927699	22522	2753	8.180893571	16.38883608
3	firesideint	0.378103361	28	10.5868941	5144	2514	2.046141607	15.59804139
74	scottmccloud	0.742708448	11	8.169792923	6758	2225	3.037303371	13.37044944
67	artinfodotcom	0.345086238	21	7.246810999	13483	2987	4.513893539	13.30317676
9	AIGAdesign	0.399021131	17	6.783359226	7889	1554	5.076576577	12.89829098
7	Tate	0.356630242	15	5.349453631	13325	1434	9.292189679	12.21685496
15	vpieters	0.544805615	17	9.261695462	2656	3173	0.837062717	11.33636343
34	hermanmiller	0.808318902	10	8.08318902	2886	1768	1.632352941	11.27973332
48	adactio	0.725068768	11	7.975756443	4189	3103	1.349983887	10.67322389
37	designmilk	0.355146246	15	5.327193694	17389	3142	5.534373011	10.39679439
31	designsponge	0.392282695	16	6.276523123	3482	3095	1.125040388	8.09594903
32	gary_hustwit	0.598228674	8	4.785829391	4796	1515	3.165676568	7.924008841
26	LACMA	0.347979187	12	4.175750248	4654	968	4.80785124	7.811407943
50	walkerartcenter	0.368418092	15	5.526271378	2060	1977	1.041982802	7.025898714
25	LightStalking	0.35559501	10	3.555950099	16991	2956	5.74797023	7.019966751
65	estria	0.508835527	9	4.579519741	2949	3188	0.925031368	5.699843212
63	eyemagazine	0.342054337	8	2.736434694	9095	2839	3.203592814	4.546048959
76	chadfowler	0.411550499	7	2.880853493	1456	2901	0.501895898	3.286497615
77	jeffpulver	0.622634131	2	1.245268263	1194	3068	0.389178618	1.383526301

Figure D.2: XRank result with 83 users from both art/design and technology topic, 200 tweets for each user, full technology vocabulary. Part 2



ID	ScreenName	DocSimilarity	WordsContribut	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
29	printmag	0.930551748	136	126.5550377	8619	2248	3.834074733	221.39639
37	designmilk	0.820564901	135	110.7762617	17389	3142	5.534373011	216.19601
9	AIGAdesign	0.916644186	121	110.9139465	7889	1554	5.076576577	210.89851
7	Tate	0.82853069	95	78.71041559	13325	1434	9.292189679	179.7555
67	artinfodotcom	0.768782756	126	96.86662727	13483	2987	4.513893539	177.82082
52	SFMOMA	0.997969051	71	70.8558026	7946	1593	4.988072819	134.01887
26	LACMA	0.783290041	89	69.71281366	4654	968	4.80785124	130.40896
0	wired	0.851091755	32	27.23493615	106202	3128	33.95204604	89.483614
62	laughingsquid	0.79261688	50	39.63084401	27642	3189	8.667920978	88.627056
71	tedtalks	0.400790228	73	29.25768666	16917	799	21.17271589	84.827289
17	johnmaeda	0.61619104	51	31.42574304	23201	1596	14.53696742	81.933458
82	Guggenheim	0.498175356	76	37.86132703	15074	1995	7.555889724	81.218082
44	timoreilly	0.952976196	28	26.68333348	57429	2982	19.25855131	75.359688
63	eyemagazine	0.753895754	55	41.46426648	9095	2839	3.203592814	68.884737
2	whitneymuseum	0.401065267	122	48.92996252	3638	2298	1.583115753	67.803675
16	google	0.829762083	24	19.91429	84295	2158	39.06163114	67.77845
25	LightStalking	0.822964556	39	32.0956177	16991	2956	5.74797023	63.361454
50	walkerartcenter	0.895063841	55	49.22851125	2060	1977	1.041982802	62.587323
72	frogdesign	0.691882184	41	28.36716953	11818	2847	4.151036178	50.796374
31	designsponge	0.9543016	39	37.2177624	3482	3095	1.125040388	48.006373
18	TEDchris	0.438161996	38	16.65015585	20172	1988	10.14688129	39.043687
10	wallpapermag	0.403836714	54	21.80718254	10372	2640	3.928787879	38.422867
32	gary_hustwit	0.519354472	43	22.33224228	4796	1515	3.165676568	36.976012
55	smithsonian	0.448799279	43	19.29836902	14277	3014	4.736894492	35.940314
19	waltmossberg	0.818999346	22	18.01798561	4451	776	5.735824742	35.547359
65	estria	0.612821249	46	28.18977743	2949	3188	0.925031368	35.086062
22	kevinrose	0.880761865	17	14.9729517	30713	3134	9.799936184	34.746239
33	Pogue	0.886008965	16	14.17614344	31427	3047	10.31407942	33.405336
12	anildash	0.749715602	24	17.99317444	12058	2753	4.379949146	32.735988
40	sacca	0.805101787	23	18.5173411	9323	2370	3.933755274	32.638455
56	SteveCase	0.914063726	16	14.62501962	23142	2875	8.049391304	31.981061
3	firesideint	0.95446198	22	20.99816355	5144	2514	2.046141607	30.937329
15	vpieters	0.567681314	44	24.97797783	2656	3173	0.837062717	30.573175
78	jennydeluxe	0.953289469	22	20.97236831	4824	2814	1.714285714	29.600707
70	zeldman	0.5255278	31	16.29136179	9504	2509	3.787963332	28.399843
6	om	0.751823318	20	15.03646635	12226	2702	4.524796447	27.622714
42	BBCClick	0.434876597	25	10.87191494	13732	1352	10.15680473	25.501501
57	khoi	0.466832846	33	15.40548392	5804	1960	2.96122449	25.034345
27	mashable	0.441643766	15	6.624656497	186144	3200	58.17	24.805442
58	espiekermann	0.647346705	20	12.9469341	12479	2505	4.981636727	24.478736
36	mikeyk	0.490628609	46	22.56891603	763	2771	0.275351859	24.372869
61	jack	0.688847222	16	11.02155555	22522	2753	8.180893571	24.219927
74	scottmccloud	0.444316989	32	14.21814364	6758	2225	3.037303371	23.269007
34	hermanmiller	0.42374079	39	16.52589082	2886	1768	1.632352941	23.061151
54	guardiantech	0.771041781	11	8.48145959	46455	3196	14.5353567	22.112223

Figure D.3: XRank result with 83 users from both art/design and technology topic, 200 tweets for each user, full art vocabulary. Part 1

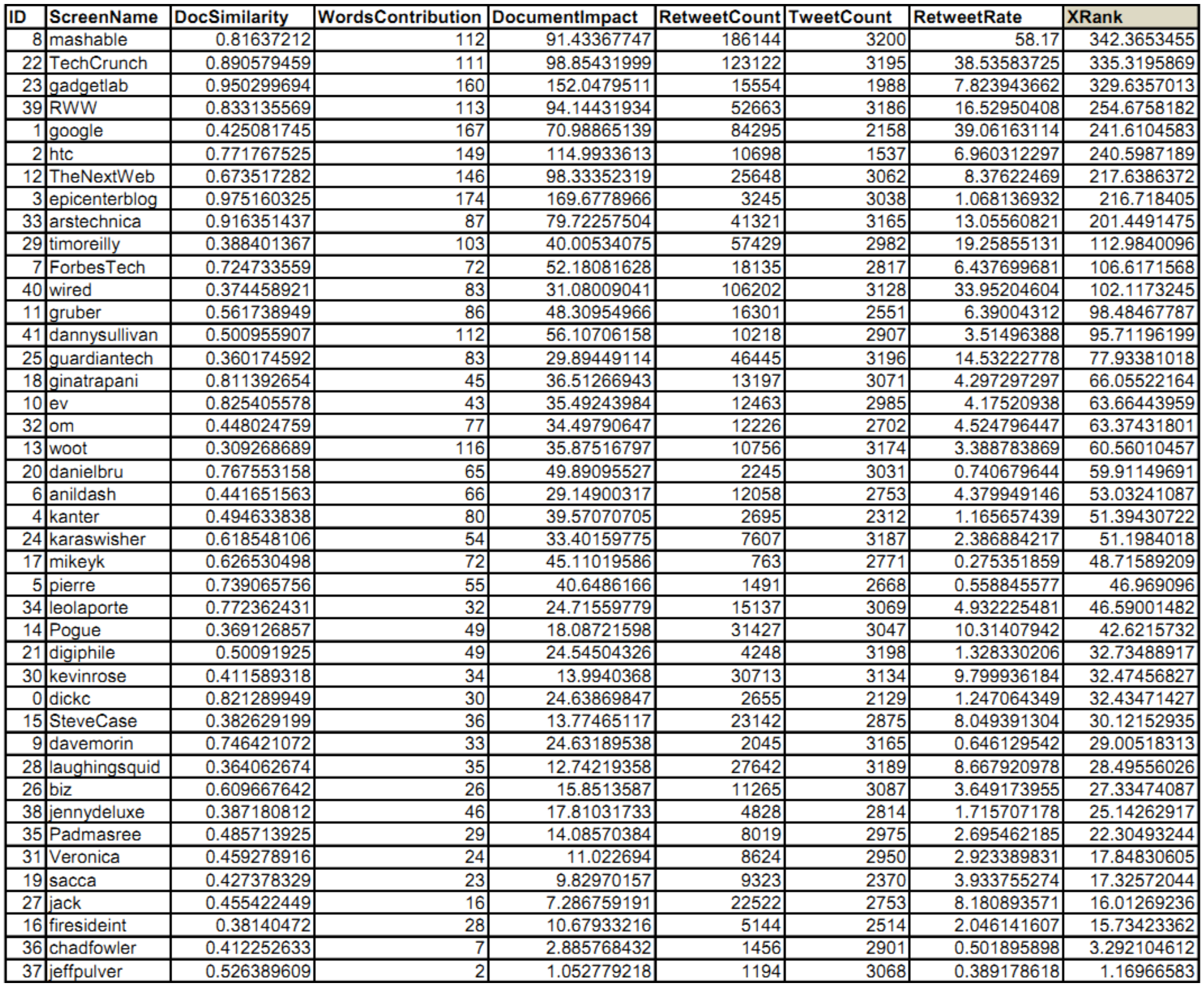


8	FCC	0.872532023	12	10.47038428	6139	962	6.381496881	21.336427
45	drawn	0.392226115	31	12.15900955	6780	2054	3.300876339	20.37202
68	NewYorker	0.408247758	19	7.756707404	36621	2529	14.48042705	20.200552
66	Veronica	0.69122325	17	11.75079524	8624	2950	2.923389831	19.027271
81	AlecJRoss	0.424907187	21	8.923050918	13134	1901	6.908995266	18.627598
47	TechCrunch	0.418392718	13	5.439105336	123122	3195	38.53583725	18.449761
28	TheNextWeb	0.488430115	16	7.814881841	25648	3062	8.37622469	17.296444
80	cshirky	0.453007385	20	9.060147705	8497	1754	4.844355758	16.986921
83	dannysullivan	0.650499029	15	9.757485436	10218	2907	3.51496388	16.645108
79	RWW	0.436945777	14	6.11724088	52663	3186	16.52950408	16.548139
75	jkottke	0.421279138	23	9.689420172	6034	1752	3.444063927	16.432489
69	arstechnica	0.425138806	14	5.951943284	41321	3165	13.05560821	15.039829
4	htc	0.421371791	17	7.16332045	10698	1537	6.960312297	14.987698
53	DellOutlet	0.443042341	24	10.63301619	2163	1400	1.545	14.653651
41	danielbru	0.431953981	28	12.09471147	2245	3031	0.740679644	14.523921
5	epicenterblog	0.38961154	28	10.90912311	3254	3038	1.071099408	13.940733
14	mezzoblue	0.462508231	24	11.10019754	2254	3167	0.711714556	13.251118
49	gadgetlab	0.391192898	14	5.476700568	15554	1988	7.823943662	11.873334
73	leolaporte	0.436995081	14	6.117931138	15137	3069	4.932225481	11.532576
24	gruber	0.547383081	10	5.473830807	16301	2551	6.39004312	11.159046
51	karaswisher	0.510748688	13	6.639732942	7607	3187	2.386884217	10.177469
77	jeffpulver	0.502262909	18	9.040732355	1194	3068	0.389178618	10.044495
39	ginatrapani	0.414485958	13	5.388317456	13197	3071	4.297297297	9.7480274
1	lessig	0.442111479	11	4.863226272	6511	1132	5.751766784	9.6026423
21	davemorin	0.4258686	19	8.091503396	2045	3165	0.646129542	9.5281152
46	digiphile	0.624555534	11	6.870110875	4248	3198	1.328330206	9.1624331
76	chadfowler	0.856726746	9	7.710540715	1456	2901	0.501895898	8.7962382
60	biz	0.465694755	10	4.656947546	11265	3087	3.649173955	8.0306336
13	Chad_Hurley	0.543236049	6	3.259416296	647	156	4.147435897	5.8350603
59	Padmasree	0.58296681	6	3.497800861	8019	2975	2.695462185	5.5388224
48	adactio	0.450879139	9	4.057912255	4189	3103	1.349983887	5.4303321
35	dickc	0.405122731	10	4.051227307	2655	2129	1.247064349	5.3330902
43	kanter	0.631501288	6	3.789007731	2695	2312	1.165657439	4.9211511
20	OpenGov	0.492642755	4	1.970571022	1646	137	12.01459854	4.8591502
64	ForbesTech	0.469778483	5	2.348892414	18135	2817	6.437699681	4.7993161
23	ev	0.407544924	6	2.445269546	12463	2985	4.17520938	4.3861937
11	pierre	0.430842487	8	3.446739896	1491	2668	0.558845577	3.9826757
38	twitter	0.398084543	3	1.194253629	31817	913	34.84884995	3.9499291
30	woot	0.403427658	5	2.017138291	10756	3174	3.388783869	3.4050881

Figure D.4: XRank result with 83 users from both art/design and technology topic, 200 tweets for each user, full art vocabulary. Part 2

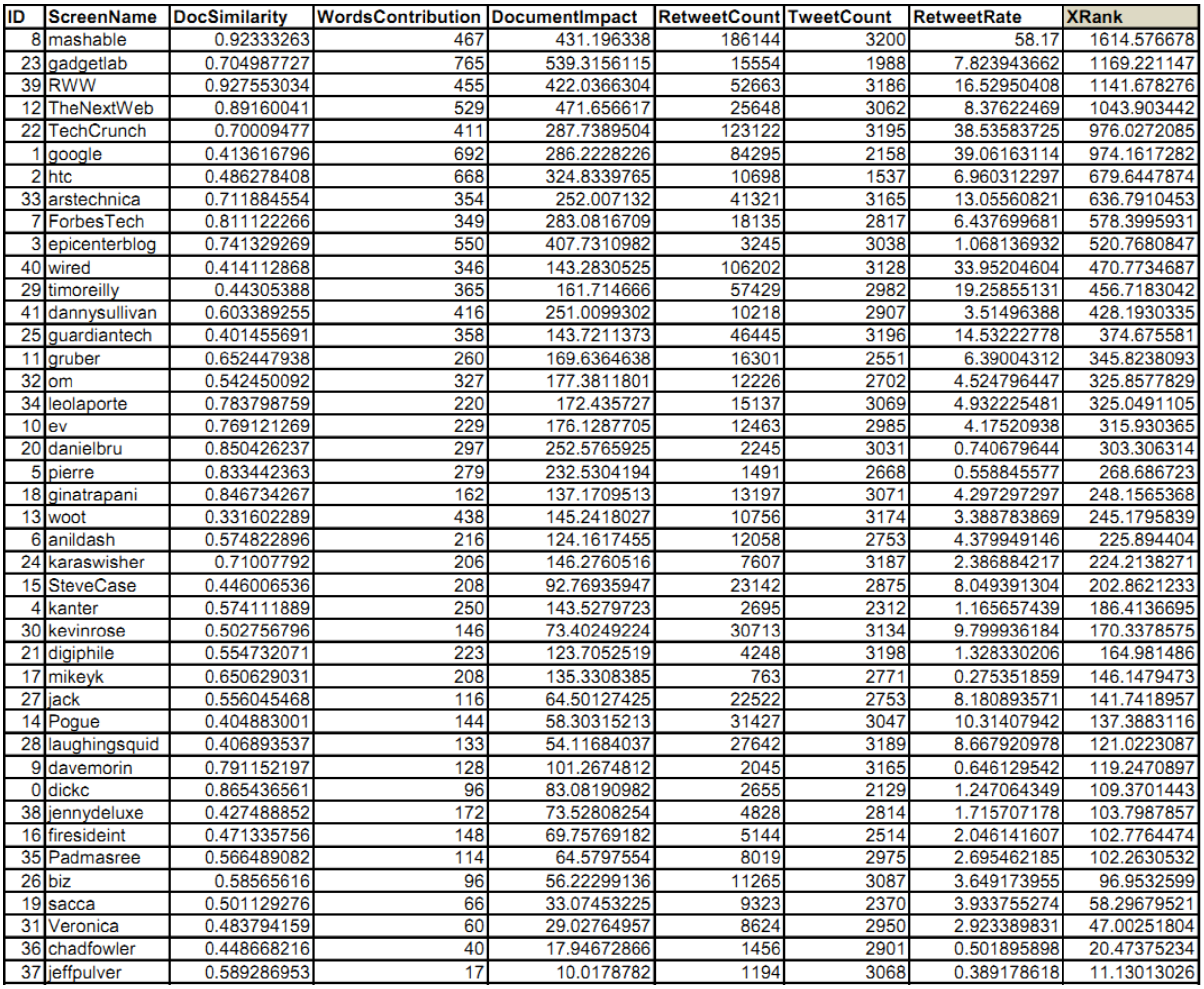
Appendix E

XRank algorithm test result based on variance of dataset size



ID	ScreenName	DocSimilarity	WordsContribution	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
8	mashable	0.81637212	112	91.43367747	186144	3200	58.17	342.3653455
22	TechCrunch	0.890579459	111	98.85431999	123122	3195	38.53583725	335.3195869
23	gadgetlab	0.950299694	160	152.0479511	15554	1988	7.823943662	329.6357013
39	RWW	0.833135569	113	94.14431934	52663	3186	16.52950408	254.6758182
1	google	0.425081745	167	70.98865139	84295	2158	39.06163114	241.6104583
2	htc	0.771767525	149	114.9933613	10698	1537	6.960312297	240.5987189
12	TheNextWeb	0.673517282	146	98.33352319	25648	3062	8.37622469	217.6386372
3	epicenterblog	0.975160325	174	169.6778966	3245	3038	1.068136932	216.718405
33	arstechnica	0.916351437	87	79.72257504	41321	3165	13.05560821	201.4491475
29	timoreilly	0.388401367	103	40.00534075	57429	2982	19.25855131	112.9840096
7	ForbesTech	0.724733559	72	52.18081628	18135	2817	6.437699681	106.6171568
40	wired	0.374458921	83	31.08009041	106202	3128	33.95204604	102.1173245
11	gruber	0.561738949	86	48.30954966	16301	2551	6.39004312	98.48467787
41	dannysullivan	0.500955907	112	56.10706158	10218	2907	3.51496388	95.71196199
25	guardiantech	0.360174592	83	29.89449114	46445	3196	14.53222778	77.93381018
18	ginatrapani	0.811392654	45	36.51266943	13197	3071	4.297297297	66.05522164
10	ev	0.825405578	43	35.49243984	12463	2985	4.17520938	63.66443959
32	om	0.448024759	77	34.49790647	12226	2702	4.524796447	63.37431801
13	woot	0.309268689	116	35.87516797	10756	3174	3.388783869	60.56010457
20	danielbru	0.767553158	65	49.89095527	2245	3031	0.740679644	59.91149691
6	anildash	0.441651563	66	29.14900317	12058	2753	4.379949146	53.03241087
4	kanter	0.494633838	80	39.57070705	2695	2312	1.165657439	51.39430722
24	karaswisher	0.618548106	54	33.40159775	7607	3187	2.386884217	51.1984018
17	mikeyk	0.626530498	72	45.11019586	763	2771	0.275351859	48.71589209
5	pierre	0.739065756	55	40.6486166	1491	2668	0.558845577	46.969096
34	leolaporte	0.772362431	32	24.71559779	15137	3069	4.932225481	46.59001482
14	Pogue	0.369126857	49	18.08721598	31427	3047	10.31407942	42.6215732
21	digiphile	0.50091925	49	24.54504326	4248	3198	1.328330206	32.73488917
30	kevinrose	0.411589318	34	13.9940368	30713	3134	9.799936184	32.47456827
0	dickc	0.821289949	30	24.63869847	2655	2129	1.247064349	32.43471427
15	SteveCase	0.382629199	36	13.77465117	23142	2875	8.049391304	30.12152935
9	davemorin	0.746421072	33	24.63189538	2045	3165	0.646129542	29.00518313
28	laughingsquid	0.364062674	35	12.74219358	27642	3189	8.667920978	28.49556026
26	biz	0.609667642	26	15.8513587	11265	3087	3.649173955	27.33474087
38	jennydeluxe	0.387180812	46	17.81031733	4828	2814	1.715707178	25.14262917
35	Padmasree	0.485713925	29	14.08570384	8019	2975	2.695462185	22.30493244
31	Veronica	0.459278916	24	11.022694	8624	2950	2.923389831	17.84830605
19	sacca	0.427378329	23	9.82970157	9323	2370	3.933755274	17.32572044
27	jack	0.455422449	16	7.286759191	22522	2753	8.180893571	16.01269236
16	firesideint	0.38140472	28	10.67933216	5144	2514	2.046141607	15.73423362
36	chadfowler	0.412252633	7	2.885768432	1456	2901	0.501895898	3.292104612
37	jeffpulver	0.526389609	2	1.052779218	1194	3068	0.389178618	1.16966583

Figure E.1: XRank result with 42 users related to technology topic, 500 tweets for each user, full technology vocabulary

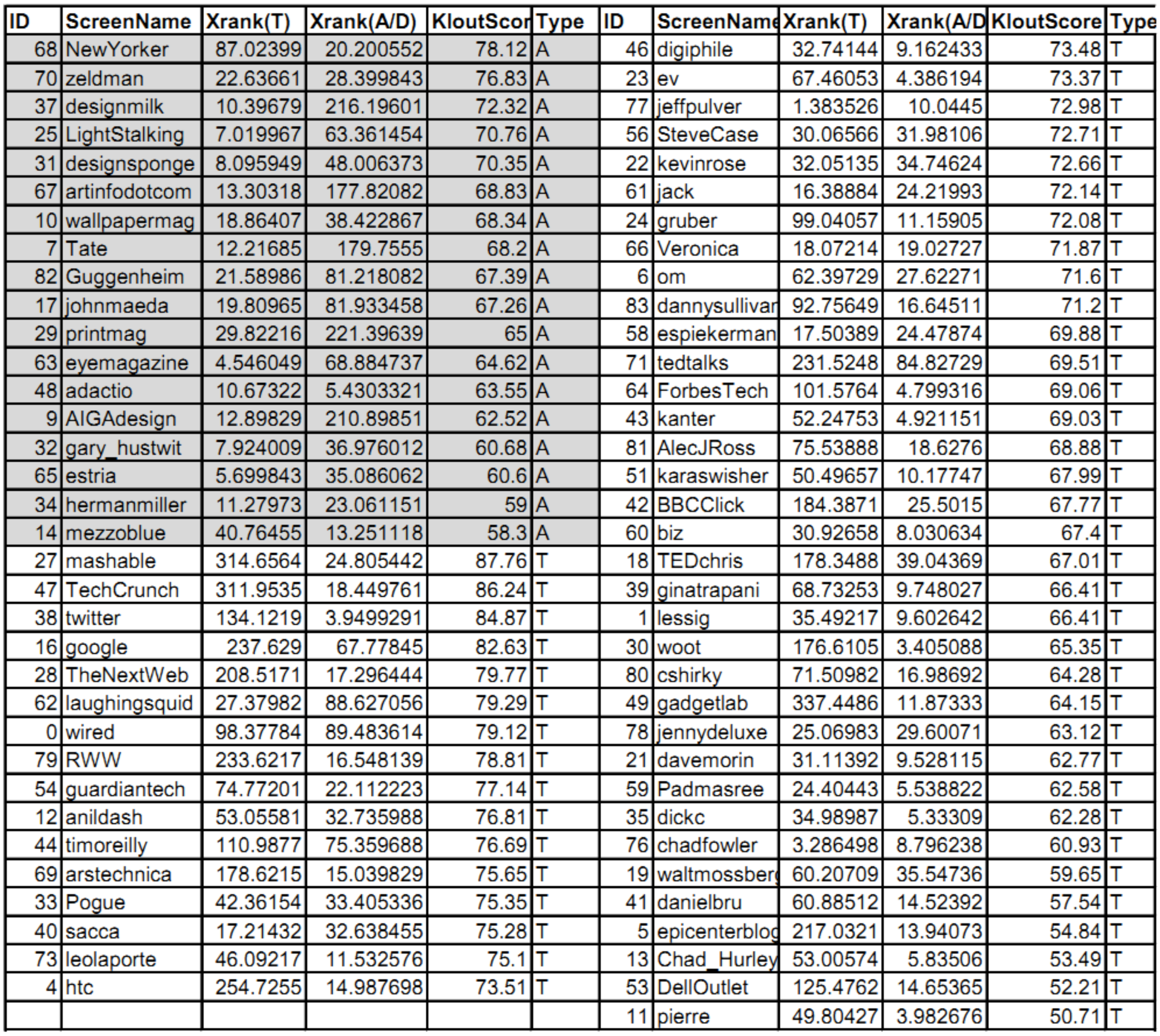


ID	ScreenName	DocSimilarity	WordsContribution	DocumentImpact	RetweetCount	TweetCount	RetweetRate	XRank
8	mashable	0.92333263	467	431.196338	186144	3200	58.17	1614.576678
23	gadgetlab	0.704987727	765	539.3156115	15554	1988	7.823943662	1169.221147
39	RWW	0.927553034	455	422.0366304	52663	3186	16.52950408	1141.678276
12	TheNextWeb	0.89160041	529	471.656617	25648	3062	8.37622469	1043.903442
22	TechCrunch	0.70009477	411	287.7389504	123122	3195	38.53583725	976.0272085
1	google	0.413616796	692	286.2228226	84295	2158	39.06163114	974.1617282
2	htc	0.486278408	668	324.8339765	10698	1537	6.960312297	679.6447874
33	arstechnica	0.711884554	354	252.007132	41321	3165	13.05560821	636.7910453
7	ForbesTech	0.811122266	349	283.0816709	18135	2817	6.437699681	578.3995931
3	epicenterblog	0.741329269	550	407.7310982	3245	3038	1.068136932	520.7680847
40	wired	0.414112868	346	143.2830525	106202	3128	33.95204604	470.7734687
29	timoreilly	0.44305388	365	161.714666	57429	2982	19.25855131	456.7183042
41	dannysullivan	0.603389255	416	251.0099302	10218	2907	3.51496388	428.1930335
25	guardiantech	0.401455691	358	143.7211373	46445	3196	14.53222778	374.675581
11	gruber	0.652447938	260	169.6364638	16301	2551	6.39004312	345.8238093
32	om	0.542450092	327	177.3811801	12226	2702	4.524796447	325.8577829
34	leolaporte	0.783798759	220	172.435727	15137	3069	4.932225481	325.0491105
10	ev	0.769121269	229	176.1287705	12463	2985	4.17520938	315.930365
20	danielbru	0.850426237	297	252.5765925	2245	3031	0.740679644	303.306314
5	pierre	0.833442363	279	232.5304194	1491	2668	0.558845577	268.686723
18	ginatrapani	0.846734267	162	137.1709513	13197	3071	4.297297297	248.1565368
13	woot	0.331602289	438	145.2418027	10756	3174	3.388783869	245.1795839
6	anildash	0.574822896	216	124.1617455	12058	2753	4.379949146	225.894404
24	karaswisher	0.71007792	206	146.2760516	7607	3187	2.386884217	224.2138271
15	SteveCase	0.446006536	208	92.76935947	23142	2875	8.049391304	202.8621233
4	kanter	0.574111889	250	143.5279723	2695	2312	1.165657439	186.4136695
30	kevinrose	0.502756796	146	73.40249224	30713	3134	9.799936184	170.3378575
21	digiphile	0.554732071	223	123.7052519	4248	3198	1.328330206	164.981486
17	mikeyk	0.650629031	208	135.3308385	763	2771	0.275351859	146.1479473
27	jack	0.556045468	116	64.50127425	22522	2753	8.180893571	141.7418957
14	Pogue	0.404883001	144	58.30315213	31427	3047	10.31407942	137.3883116
28	laughingsquid	0.406893537	133	54.11684037	27642	3189	8.667920978	121.0223087
9	davemorin	0.791152197	128	101.2674812	2045	3165	0.646129542	119.2470897
0	dickc	0.865436561	96	83.08190982	2655	2129	1.247064349	109.3701443
38	jennydeluxe	0.427488852	172	73.52808254	4828	2814	1.715707178	103.7987857
16	firesideint	0.471335756	148	69.75769182	5144	2514	2.046141607	102.7764474
35	Padmasree	0.566489082	114	64.5797554	8019	2975	2.695462185	102.2630532
26	biz	0.58565616	96	56.22299136	11265	3087	3.649173955	96.9532599
19	sacca	0.501129276	66	33.07453225	9323	2370	3.933755274	58.29679521
31	Veronica	0.483794159	60	29.02764957	8624	2950	2.923389831	47.00251804
36	chadfowler	0.448668216	40	17.94672866	1456	2901	0.501895898	20.47375234
37	jeffpulver	0.589286953	17	10.0178782	1194	3068	0.389178618	11.13013026

Figure E.2: XRank result with 42 users related to technology topic, 800 tweets for each user, full technology vocabulary

Appendix F

Comparison between XRank and Klout Score

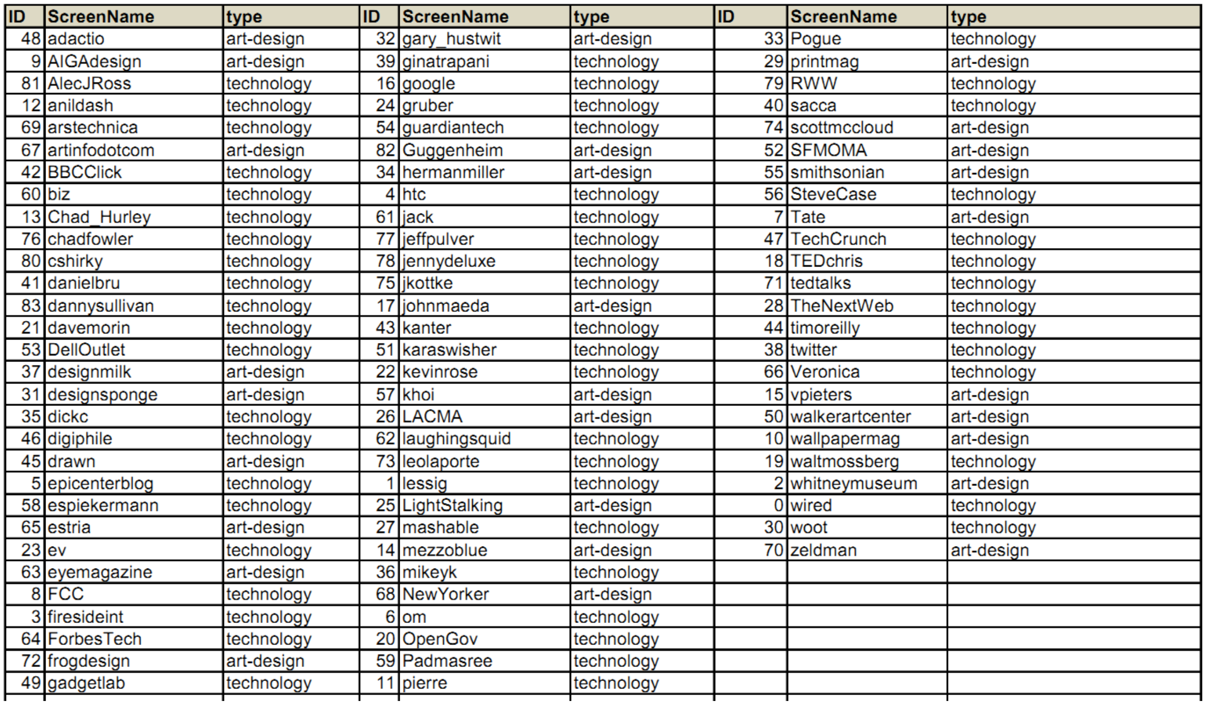


ID	ScreenName	Xrank(T)	Xrank(A/D)	KloutScor	Type
68	NewYorker	87.02399	20.200552	78.12	A
70	zeldman	22.63661	28.399843	76.83	A
37	designmilk	10.39679	216.19601	72.32	A
25	LightStalking	7.019967	63.361454	70.76	A
31	designsponge	8.095949	48.006373	70.35	A
67	artinfodotcom	13.30318	177.82082	68.83	A
10	wallpapermag	18.86407	38.422867	68.34	A
7	Tate	12.21685	179.7555	68.2	A
82	Guggenheim	21.58986	81.218082	67.39	A
17	johnmaeda	19.80965	81.933458	67.26	A
29	printmag	29.82216	221.39639	65	A
63	eyemagazine	4.546049	68.884737	64.62	A
48	adactio	10.67322	5.4303321	63.55	A
9	AIGAdesign	12.89829	210.89851	62.52	A
32	gary_hustwit	7.924009	36.976012	60.68	A
65	estria	5.699843	35.086062	60.6	A
34	hermanmiller	11.27973	23.061151	59	A
14	mezzoblue	40.76455	13.251118	58.3	A
27	mashable	314.6564	24.805442	87.76	T
47	TechCrunch	311.9535	18.449761	86.24	T
38	twitter	134.1219	3.9499291	84.87	T
16	google	237.629	67.77845	82.63	T
28	TheNextWeb	208.5171	17.296444	79.77	T
62	laughingsquid	27.37982	88.627056	79.29	T
0	wired	98.37784	89.483614	79.12	T
79	RWW	233.6217	16.548139	78.81	T
54	guardiantech	74.77201	22.112223	77.14	T
12	anildash	53.05581	32.735988	76.81	T
44	timoreilly	110.9877	75.359688	76.69	T
69	arstechnica	178.6215	15.039829	75.65	T
33	Pogue	42.36154	33.405336	75.35	T
40	sacca	17.21432	32.638455	75.28	T
73	leolaporte	46.09217	11.532576	75.1	T
4	htc	254.7255	14.987698	73.51	T
46	digiphile	32.74144	9.162433	73.48	T
23	ev	67.46053	4.386194	73.37	T
77	jeffpulver	1.383526	10.0445	72.98	T
56	SteveCase	30.06566	31.98106	72.71	T
22	kevinrose	32.05135	34.74624	72.66	T
61	jack	16.38884	24.21993	72.14	T
24	gruber	99.04057	11.15905	72.08	T
66	Veronica	18.07214	19.02727	71.87	T
6	om	62.39729	27.62271	71.6	T
83	dannysullivar	92.75649	16.64511	71.2	T
58	espiekerman	17.50389	24.47874	69.88	T
71	tedtalks	231.5248	84.82729	69.51	T
64	ForbesTech	101.5764	4.799316	69.06	T
43	kanter	52.24753	4.921151	69.03	T
81	AlecJRoss	75.53888	18.6276	68.88	T
51	karaswisher	50.49657	10.17747	67.99	T
42	BBCClick	184.3871	25.5015	67.77	T
60	biz	30.92658	8.030634	67.4	T
18	TEDchris	178.3488	39.04369	67.01	T
39	ginatrapani	68.73253	9.748027	66.41	T
1	lessig	35.49217	9.602642	66.41	T
30	woot	176.6105	3.405088	65.35	T
80	cshirky	71.50982	16.98692	64.28	T
49	gadgetlab	337.4486	11.87333	64.15	T
78	jennydeluxe	25.06983	29.60071	63.12	T
21	davemorin	31.11392	9.528115	62.77	T
59	Padmasree	24.40443	5.538822	62.58	T
35	dickc	34.98987	5.33309	62.28	T
76	chadfowler	3.286498	8.796238	60.93	T
19	waltmossber	60.20709	35.54736	59.65	T
41	danielbru	60.88512	14.52392	57.54	T
5	epicenterblog	217.0321	13.94073	54.84	T
13	Chad_Hurley	53.00574	5.83506	53.49	T
53	DellOutlet	125.4762	14.65365	52.21	T
11	pierre	49.80427	3.982676	50.71	T

Figure F.1: XRank and Klout Score for 83 users, 200 tweets for each user, full size vocabularies

Appendix G

Twitter user type from twitter suggestion list in test dataset



ID	ScreenName	type	ID	ScreenName	type	ID	ScreenName	type
48	adactio	art-design	32	gary_hustwit	art-design	33	Pogue	technology
9	AIGAdesign	art-design	39	ginatrapani	technology	29	printmag	art-design
81	AlecJRoss	technology	16	google	technology	79	RWW	technology
12	anildash	technology	24	gruber	technology	40	sacca	technology
69	arstechnica	technology	54	guardiantech	technology	74	scottmccloud	art-design
67	artinfodotcom	art-design	82	Guggenheim	art-design	52	SFMOMA	art-design
42	BBCClick	technology	34	hermanmiller	art-design	55	smithsonian	art-design
60	biz	technology	4	htc	technology	56	SteveCase	technology
13	Chad_Hurley	technology	61	jack	technology	7	Tate	art-design
76	chadfowler	technology	77	jeffpulver	technology	47	TechCrunch	technology
80	cshirky	technology	78	jennydeluxe	technology	18	TEDchris	technology
41	danielbru	technology	75	jkottke	technology	71	tedtalks	technology
83	dannysullivan	technology	17	johnmaeda	art-design	28	TheNextWeb	technology
21	davemorin	technology	43	kanter	technology	44	timoreilly	technology
53	DellOutlet	technology	51	karaswisher	technology	38	twitter	technology
37	designmilk	art-design	22	kevinrose	technology	66	Veronica	technology
31	designsponge	art-design	57	khoi	art-design	15	vpieters	art-design
35	dickc	technology	26	LACMA	art-design	50	walkerartcenter	art-design
46	digiphile	technology	62	laughingsquid	technology	10	wallpapermag	art-design
45	drawn	art-design	73	leolaporte	technology	19	waltmossberg	technology
5	epicenterblog	technology	1	lessig	technology	2	whitneymuseum	art-design
58	espiekermann	technology	25	LightStalking	art-design	0	wired	technology
65	estria	art-design	27	mashable	technology	30	woot	technology
23	ev	technology	14	mezzoblue	art-design	70	zeldman	art-design
63	eyemagazine	art-design	36	mikeyk	technology			
8	FCC	technology	68	NewYorker	art-design			
3	firesideint	technology	6	om	technology			
64	ForbesTech	technology	20	OpenGov	technology			
72	frogdesign	art-design	59	Padmasree	technology			
49	gadgetlab	technology	11	pierre	technology			

Figure G.1: This figure shows topics which all 83 users that used in the experiment belong to

Appendix H

Vocabulary on technology topic

Full size of technology vocabulary
Ericsson google Slate LG iphone API APIs Code Twitter PS3 Android developers Device Incredible Tablet applications Cisco Verizon Satellite Honeycomb phone technology 3GS Inspire Xbox legend Panasonic htc nano diamond2 ted telecom cloud labs wifi gmail chip mac vodafone buzz translate pro2 xperia wwdc webcast ipad pc geo apple hardware crack adobe webos kindle smartphone digital carriers wildfire Social Media network Microsoft cameras gallery computer Sense spotify voip ios nexus dropbox tablets 3DS blackberry adsense podcast antenna fcc SD IBM mobile itunes battery skype mwc a4 youtube facetime ipod yahoo zune gsm googleio xoom chrome Netbook Amazon Canon Hulu keyboard MacBook NFC jailbreak LTE Disk Screen 3D Motorola 4G G2 G1 McAfee Hero laptop hp HD wireless Flyer Opera Samsung linux Bluetooth iMac 16GB 32G iPhone4 sony GPS Zynga ARM

Half size of technology vocabulary
Ericsson google Slate LG iphone API APIs Code Twitter PS3 Android developers Device Incredible Tablet applications Cisco Verizon Satellite Honeycomb phone technology 3GS Inspire Xbox legend Panasonic htc nano diamond2 ted telecom cloud labs wifi gmail chip mac vodafone buzz translate pro2 xperia wwdc webcast ipad pc geo apple hardware crack adobe webos kindle smartphone digital carriers wildfire Social Media network Microsoft cameras gallery

Technology vocabulary consists of nine words
Ericsson google LG Cisco Verizon Honeycomb legend Panasonic htc

Appendix I

Vocabulary on art topic

music art design beauty webfonts Beautiful feeling petapixel pixel theme conceptual life magazines wordpress panels choice awesome artist vision IKEA Vienna favorite gorgeous sense imagine Redesign interface view stylist stylish Designer Designed magic Flash designs scene Auction Advertising advertise color patterns inspiration Photograph musical architectural pink typographica brown bold paintings photoartgallery cartoonist Graphics retrospective dresser museum Geneva fantastic amazing Sculpture sunflower carving photographyelf Studio designblahg images print paint letterpress Classic font solid jewelry pet map cartoonists Imprint inspiring photographs romantic Louvre elegant tnycloseread Ivy prints picture impressive artworks wood cabinet canvas sculptural Arial style grace essential showcase Light necklace Arts Landscape vintage Avatar concerts Writer whitney

Bibliography

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, G. Mishne, E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media with an application to community-based question answering," in *In Proceedings of WSDM*, 2008.

[2] N. O. Andrews and E. A. Fox, "Recent developments in document clustering," Tech. Rep., 2007.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 12, pp. 43–77, 1994.

[4] M. W. Berry, Z. Drma, Elizabeth, and R. Jessup, "Matrices, vector spaces, and information retrieval," *SIAM Review*, vol. 41, pp. 335–362, 1999.

[5] A. Bjorck, "Numerical methods for least squares problems," *Mathematics of Computation*, 1996.

[6] K. P. Bube, "Seismic traveltime tomography," 1998.

[7] Barracuda labs annual report 2009. Business Insider. [Online]. Available: http://barracudalabs.com/downloads/BarracudaLabs2009AnnualReport-FINAL.pdf

[8] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in *in ICWSM' 10: Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[9] R. C. Chen, I. Y. Lee, Y. C. Lee, and Y. L. Lo, "Upgrading domain ontology based on latent semantic analysis and group center similarity calculation," *IEEE*, vol. 1-4244-2384-2, pp. 1495–1500, 2010.

[10] 5 ways twitter changed how we communicate. CNN. [Online]. Available: http://articles.cnn.com/2011-03-21/tech/twitter.birthday.communication_1_twitter-microblogging-celebrities?_s=PM:TECH

[11] M. L. COUNCIL, "Leveraging social networking sites in marketing communications," Tech. Rep.

[12] 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*, vol. 41, no. 6, pp. 391–407, 1990.

[13] W. B. Frakes and R. A. Baeza-Yates, Eds., *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992.

[14] W. N. Gansterer, A. G. K. Janecek, and R. Neumayer, "Spam filtering based on latent semantic indexing," Tech. Rep., 2007.

[15] G. Golub, V. Klema, and G. W. Stewart, "Rank degeneracy and least squares problems," Tech. Rep., 1976.

[16] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust." ACM Press, 2004, pp. 403–412.

[17] J. Humpolicek, "Text document classification," Tech. Rep., 2005.

[18] What is social media? icrossing. [Online]. Available: http://www.icrossing.co.uk/fileadmin/uploads/eBooks/

[19] iProspect, "Social networking user behaviour study," Tech. Rep., 2007.

[20] Reviews from epinions. Klout Dev. [Online]. Available: http://epinions.com/

[21] Understanding the influence metric:what's a klout score? Klout Dev. [Online]. Available: http://klout.com/kscore

[22] W. Kraaij and R. E. Pohlmann, "Viewing stemming as recall enhancement," in *In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996, pp. 40–48.

[23] R. Krovetz, "Viewing morphology as an inference process," in *Research and Development in Information Retrieval*, 1993, pp. 191–202.

[24] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, 1998.

[25] R. T.-W. Lo, B. He, and I. Ounis, "Automatically building a stopword list for an information retrieval system," *5th Dutch-Belgium Information Retrieval Workshop*, 2005.

[26] P. Melville, V. Sindhwani, R. D. Lawrence, E. Meliksetian, Y. Liu, P.-Y. Hsueh, and C. Perlich, "Machine learning for social media analytics."

[27] J. L. A. D. W. Myers, *Research Design and Statistical Analysis (second edition ed.)*. Lawrence Erlbaum, 2003.

[28] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall, 1978.

[29] E. M. Rogers, *Diffusion of innovations*. New York: Free Press, 1995.

[30] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman, “Influence and passivity in social media,” *CoRR*, vol. abs/1008.1253, 2010.

[31] P. Ruch, R. Baud, and M. Hilario, “Text mining and information retrieval in medical records: an inquiry into automatic spelling correction,” 2002.

[32] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Communications of the ACM,*, vol. 41, pp. 613–620, 1990.

[33] A. P. Saygin, I. Cicekli, and V. Akman, “Turing test: 50 years later,” *Minds and Machines*, vol. 10, pp. 463–518, 2000.

[34] J. A. Scales, P. Docherty, and A. Gersztenkorn, “Regularisation of nonlinear inverse problems: imaging the near-surface weathering layer,” *Inverse Problems*, vol. 6, no. 1, p. 115, 1990. [Online]. Available: http://stacks.iop.org/0266-5611/6/i=1/a=011

[35] M. Schreiner, M. B. W. Wolfe, D. Laham, T. K. L, T. K. Landauer, W. Kintsch, B. Rehder, and B. Rehder, “Using latent semantic analysis to assess knowledge: Some technical considerations,” 1998.

[36] A. M. Turing, “Computing machinery and intelligence,” *Mind*, vol. 49, pp. 433–460, 1950.

[37] numbers. Twitter Blog. [Online]. Available: http://blog.twitter.com/2011/03/numbers.html

[38] Boolean retrieval model and controlled vocabulary techniques. University of Maryland. [Online]. Available: http://www.coli.uni-saarland.de/~schulte/Teaching/Klassifikation-04/

[39] C. Ward, “Word of the web, guidelines for advertisers,” Tech. Rep., 2007.

[40] M. B. W. Wolfe, M. E. S. B. Rehder, D. Laham, P. W. Foltz, W. Kintsch, and T. K. L. , “Learning from text: Matching readers and texts by latent semantic analysis,” *Discourse Processes*, vol. 25, pp. 309–336, 1998.

[41] J. Zhang, M. S. Ackerman, and L. Adamic, “Expertise networks in online communities: structure and algorithms,” in *WWW ’07: Proceedings of the 16th international conference on World Wide Web*. ACM Press, 2007, pp. 221–230.

[42] X. Zhu, “Semi-supervised learning literature survey,” Tech. Rep., 2008.

[43] F. Zou, F. L. Wang, X. Deng, and S. Han, “Evaluation of stop word lists in chinese language,” Tech. Rep., 2007.