A Generalized Framework for Quantifying Trust of Social Media Text Documents

Tuhin Sharma
Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee
Roorkee, India
tuhinsharma121@gmail.com

Durga Toshniwal
Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee
Roorkee, India
durgatoshniwal@gmail.com

***Abstract*—Social media has become a very popular place for users seeking knowledge about a wide variety of topics. While it contains many helpful documents, it also contains many useless and malicious documents or spams. For a casual observer it is very hard to identify high quality or trustworthy documents. As the volume of such data increases the task for identifying high quality or trustworthy documents becomes more and more difficult. A huge number of research works have focused on quantifying trust in certain specific social network domain. Some have quantified trust based on social graph with relationships. In this work, we use such social graph named Reduced node Social Graph with Relationships (RSGR) and we develop a three-step syntax and semantic based trust mining framework. Here we generalize the concept of trust mining for all structured as well as unstructured unsupervised text documents from all social network domains. We calculate trust based on metadata, trust based on relationships with other documents and finally we propagate the trust calculated so far along various relationship edges accordingly to calculate the final trust.**

***Keywords*-Relationship mining, Trust Mining, Social graph.**

I. Introduction

Social Media is growing day by day at an increasing rate of growth of millions of documents per day. A document could be a facebook post, a tweet, a blog, a review or even a video. A substantial portion of these documents is useful and is an excellent source for performing various social media analysis like sentiment analysis, segmentation analysis, etc. to obtain knowledge. But due to the availability of this huge amount of information, there is an important need for differentiating between good and bad documents, since all these documents are not useful unless manually read. In the current scenario of social media analysis, documents are fed individually. But, as there is no relationship between the documents, we have knowledge of every document individually but not in presence of other documents. So, we cannot say anything about the reliability of the individual documents. Let us call the reliability as Trust. Hence the trust of the data is unknown and there is absolutely no quick method to diagnose if the document could be actually trusted. The need for Trust Mining comes from the fact that the better understanding of the data we have, the better will be the analysis of that data. So, before feeding the documents into social media analysis, some trust distribution should be assigned to the dataset so that low trustworthy documents can be filtered out for the improvement of the analysis.

In this paper, we use Reduced node Social Graph with Relationships (RSGR)[14]. On the basis of the RSGR and various information about the document, such as information about Author, Domain etc. we develop a three-step syntax and semantic based Trust Mining approach. The novelty of our work is as follows. First we develop a Metadata Trust Score Algorithm (MTSA) to calculate trust based on metadata. In the second step we develop a Relative Trust Score Algorithm (RTSA) to calculate trust in the presence of other documents. Finally we develop a Propagated Trust Score Algorithm (PTSA) to propagate trust through various relationship edges.

II. Related Work

Earlier research work on trust mining from social media data normally focus on trust network having explicit trust. Golbeck et al. propose a method for the application of social media analysis on multi-dimensional networks evolved from ontological trust specifications[1]. Guha et al. develop a trust as well as a distrust propagation model[2]. These approaches rely only on the structure of the web of trust. So, the accuracy cannot be guaranteed as there is lack of context and content such as, documents' topics, users' activities etc. To overcome this problem Christian Bizer et al. propose the usage of context and content based trust mechanisms to develop a trust architecture which uses a combination of reputation, context and content-based trust mechanisms[3].

Agichtein et al.[4] propose a classification framework for combining evidence from different information sources. Blumenstock propose a method that simply uses the word count of the articles as a measure of quality of Wikipedia[5]. Zolfaghar et al. develop a system consisting of various classification models such as support vector machine, decision tree etc. to predict trust and distrust relations [6]. But they do not pay any attention to mine information from the unstructured data. McGuinness et al. measure trust based on the frequency of occurrences of the encyclopedia terms in Wikipedia articles [7].

Lim et al. propose a trust antecedent framework which considers ability, integrity and benevolence as the three key factors for the formulation of trust[8][9]. But the problem is that it is very difficult to acquire all the three key factors in online communities. Yu Zhang and Tong Yu uses semantic-based trust mining mechanism to build domain ontology[11]. But they only focus on structured data. Sai T. Moturu and Huan Liu develop an unsupervised approach to quantify trustworthiness[13]. They first identify the features and then quantify the trust. But they do not calculate trust based on relationships in social network. Matsuo and Yamamoto use features extracted from product reviews, user profiles and existing trust relations to predict trust between users[10].

In our previous work[14] we propose a syntax and semantic based relationship mining approach for establishing relationships between social media text documents irrespective of the type of document and the source domain. As a result we finally get a Reduced node Social Graph with relationships (RSGR). In the current work we use this RSGR for quantifying trust of social media documents. The next sections describe the work we have done. In section III the proposed work is elaborately discussed. In section IV experimental results are shown.

III. Proposed Work

The framework for the proposed work is shown in Fig.1. Feature extraction and Finding different types of relationships are briefly depicted in III-A and III-B respectively as per[14]. We mainly concentrate on III-C which focuses on the trust mining approach.

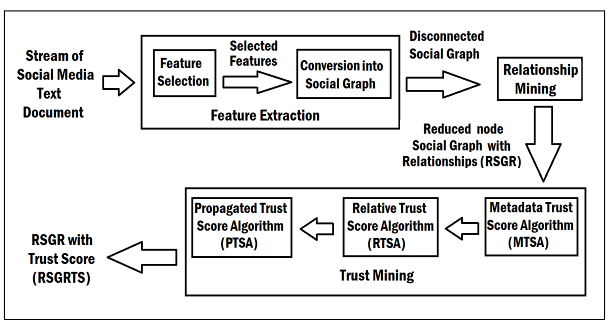


Figure 1: Proposed framework for Trust mining

A. Feature extraction

Each of the incoming documents can be categorized into 4 types which are Boards (e.g. facebook posts), Reviews (e.g. tripadvisor review), Microblogs (e.g. twits) and Videos (e.g. youtube videos). For every document we extract information regarding its Author, Timestamp, Domain and Text and we create WHO,WHEN,WHERE and DOCUMENT nodes respectively and store them accordingly in graph structure as shown in Fig.2. We additionally create WHAT nodes to store different concepts or keywords of text documents and to store interest of Authors. For individual documents different edges are also shown in Fig.2. The edges shown are not assigned any weight.

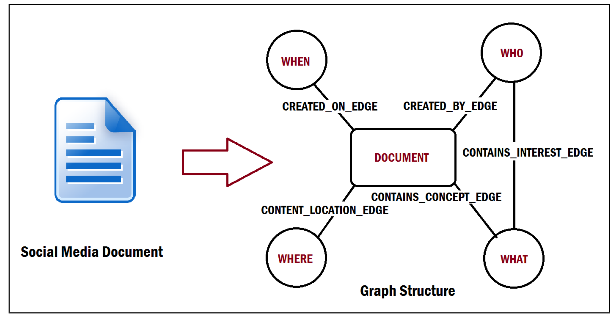


Figure 2: Each document converted into graph structure [14]

DOCUMENT node keeps track of the actual document. It contains the *Url*, *Type* of the document, *Texthtml, Subjecthtml, Review rating* etc.

WHO node contains information about the Author like *Real name, Username, Location, Gender, Profile-url,Author description* etc.

WHERE node contains information about where the document has been created like *Domain, Geo-location* etc. Additionally we store the Pagerank score of respective domains as per Quantcast dataset[18]. It is calculated as the difference between the pagerank of the domain and the highest possible rank in the Quantcast dataset.

WHEN node contains information about timestamp like *date of creation*, *date of registration* etc.

WHAT nodes represent concepts i.e. important keywords which solely determine the meaning of a document or the interest of an Author. The concepts fetched from *Texthtml* and *Subjecthtml* are associated with corresponding DOCUMENT node and those fetched from *Author description* are associated with respective WHO node. From corresponding fields plain text data is extracted and tagged using opennlp POS tagger[15]. Then the plain text is tokenized and stopwords and punctuations are removed. Here we store top 10 concepts out of which at most 3 concepts will be proper nouns according to the frequency of occurrence in respective documents. If there exist two different concepts with same term frequency then we select the concept which occurred first in the text. All the concepts are stemmed and both stemmed and the actual concepts are stored in the corresponding WHAT nodes.

B. Relationship Mining

Depending on the type of nodes different approach is followed to remove duplicates and to establish relationships.

1) Relationship between WHO nodes:

a) Common interest relationship: It represents whether two Authors have any similar interests. To mine this relation the WHAT nodes associated with one WHO node are compared against those associated with the other WHO node using Wordnet dictionary and Freebase. While comparing 2 WHAT nodes if both are Proper nouns then we calculate Wu-Palmer similarity(WPS)[12] value according to freebase, otherwise if both of them are not Proper nouns then we compare them using Wordnet dictionary. Let, for two WHO nodes m is the number of the pairs of the Proper nouns those are similar and n is the number of other similar concepts. Then we define the $match\_value$ and common interest similarity value ($CISV$) as expressed in Eq.1 and Eq.2. avg_length is calculated by dividing the total no. of concepts associated with those two WHO nodes by 2.

$$match\_value = \sum_{i=1}^{m} WPS_i + n \tag{1}$$

$$CISV = match\_value/avg\_length \tag{2}$$

If $CISV$ is greater than 0.5 the two WHO nodes are connected by a COMMON_INTEREST_SIMILARITY_EDGE (CISE) in which the $CISV$ is stored.

b) User-id similarity relationship.: It represents whether two Authors are similar or not on the basis of their user-id. *Username*(**U**), *Gender*(**G**), *Description*(**D**), *Location*(**L**) and *Original name*(**O**) are compared. For *Username* Jaro-Winkler coefficient is used. For *Description* the $CISV$ is used. For rest of them Jaccard coefficient is used. So for each of the attributes we get respective similarity values (S_i). The priorities of the similarity of these 5 fields in descending order are as follows:-

G > **L** > **D** > **O** > **U**

Let, two WHO nodes have different gender then irrespective of other fields they cannot be same person. If gender is same for two WHO nodes but they belong to two different countries then they cannot be the same person and so on. So, the similarity due to Gender should get highest weightage and that due to Username should get the lowest weightage when we use those similarity values to calculate the user-id similarity value. Often we find that some attributes are applicable to certain documents and some are not. For a facebook document we shall find a attribute named *number of friends* but in case of trip-advisor document we cannot find such field. So, while calculating the similarity we shall consider only those attributes for which the values are present in both of the documents. Let, for m attributes $A_1, A_2 ... A_m$ values are present in both WHO nodes. The priorities in descending order are as shown in Eq.3.

$$A_1 > A_2 > .... > A_m [1 \leq m \leq 5] \tag{3}$$

We assign weights W_i to respective S_i as per Eq.4.

$$W_i = \frac{m - i + 1}{\sum_{i=1}^{m} i} \tag{4}$$

User-id similarity value (USV) is calculated as per Eq.5.

$$USV = \sum_{i=1}^{m} W_i S_i \tag{5}$$

If USV is greater than 0.8 the two WHO nodes are connected by a USERID_SIMILARITY_EDGE (USE) edge in which the USV is stored.

2) Relationship between DOCUMENT and WHERE: We analyze the *Texthtml* of the DOCUMENT and find out different domain names and connect those corresponding WHERE nodes with the DOCUMENT node via PAGE_REFERENCE_EDGE (PRE).

3) Relationship between DOCUMENT node:

a) Document Concept similarity relationship: We connect two DOCUMENT nodes by a DOCUMENT_CONCEPT_SIMILARITY_EDGE (DCSE) if they have any similar concept. To mine this relation same procedure is followed as the process adopted while establishing CISE between WHO nodes.

b) Document Reference relationship: It represents if a DOCUMENT is referred by another DOCUMENT. To find this relationship we analyze *Texthtml* to find out if there exists any link to other document. If we find any such document then we connect those documents using DOCUMENT_REFERENCE_EDGE (DRE).

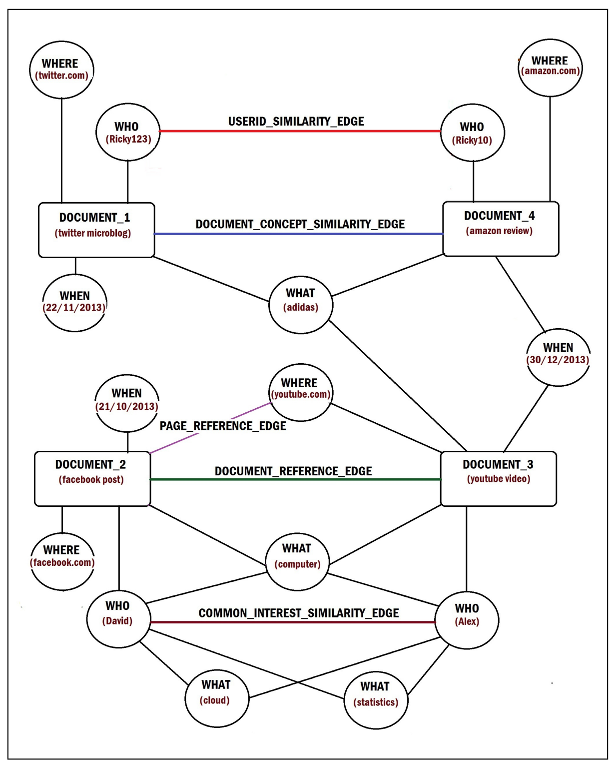


Figure 3: RSGR [14]

Finally we get Reduced node Social Graph with Relationships (RSGR) after the relationship mining as shown in Fig.3.We shall use this RSGR graph for trust mining.

C. Trust mining

We calculate the trust in 3 steps. At every step we consider different basis for computing trust. These are as follows:-

1) Trust based on metadata: So far we have stored useful attributes for each type of nodes. But the importance of every attributes are not same. For example, some attributes are used in querying the RSGR, some are used in Relationship Mining and some are used in Trust Mining. Some attributes are used in all of them. Also there are some attributes without which corresponding nodes do not make any sense. For example, *Url* must be present in node of type DOCUMENT. So we assign some weights to each attribute depending on whether they are used in Graph Query (**GQ**), Relationship Mining (**RM**), Trust Mining(**TM**) and also depending on a Static Priority (**SP**). The Metadata Trust Score (MTS) is calculated by adding these associated weights only if the corresponding attributes are present. In this step trust is assigned only to WHO, WHERE and DOCUMENT nodes in a scale of $[0, 0.2]$.

Let, $X_1, X_2...X_m$ are the attributes present in a node where m is the total number of attributes present in it. Let, F_j be the usage frequency having value $j \in \{1, 2, 3, 4\}$ corresponding to X_i. Suppose, S_j represents support for F_j i.e. for S_j . We develop Metadata Trust Score Algorithm (MTSA) to calculate MTS of a node as discussed in Algorithm 1. Here, $f = 0.2$ is the scaling factor.

Algorithm 1 Metadata Trust Score Agorithm (MTSA)

INPUT: Set of attributes $X_1, X_2...X_m$ of a node N
OUTPUT: MTS of N

Initialize $MTS = 0$
for all X_i **do**
  Calculate F_j by summing the number of times X_i is used in **GQ, RM, TM** and **SP**
end for
for all F_j **do**
  Calculate S_j by counting the number of attributes having usage frequency is F_j
end for
for all X_i **do**
  $W_i = \frac{F_j S_j}{\sum_{i=1}^{4} F_j S_j} \times \frac{1}{F_j} \times f = \frac{F_j}{\sum_{i=1}^{4} F_j S_j} \times f$
end for
for all X_i **do**
  if X_i is not null **then**
    $MTS = MTS + W_i$
  end if
end for

One example for calculating W_i for document type *Reviews* and node type DOCUMENT is shown in Table 1,

2) Trust Based on Relationship: The relative trust represents the reliability of documents in the vicinity of other documents. We select deciding numeric attributes depending on the type of document and node type so that we can calculate relative trust score (RTS) based on them. Here we develop Relative Trust Score Algorithm (RTSA) to find

Table I: Calculation of W_i using MTSA

Attribute		**GQ**	**RM**	**TM**	**SP**	F_j	S_j	W_i
Id(X_i)	**Name**							
X_1	Postsize	✓		✓		2	1	0.0307
X_2	Texthtml	✓	✓	✓	✓	4	2	0.0615
X_3	Titlehtml	✓	✓	✓	✓			0.0615
X_4	Url	✓	✓		✓	3	1	0.0461

RTS of WHO, WHERE and DOCUMENT nodes. We use Z-score for this purpose. The z-score is a measure of how many units of standard deviation the raw attribute value is from the mean value. Thus, the z-score is a relative measure instead of an absolute measure. Then we define Dispersion Score (DS) for each numeric attribute values from respective attributes and finally find Relative Trust Score (RTS).

The RTSA for document type D and node type N is described in Algorithm 2. Here, $A^1, A^2...A^m$ are the m deciding attributes present in n number of nodes $N_1, N_2...N_n$ of type N. A^i_j represents the numeric value of A^i for N_j, $[1 \leq i \leq m, 1 \leq j \leq n]$. $Z(A^i_j)$ and $DS(A^i_j)$ represents Z-score and Dispersion score of A^i_j respectively. Clearly, $0 \leq DS(A^i_j), RTS(N_j) \leq 1$.

Algorithm 2 Relative Trust Score Agorithm (RTSA)

INPUT: $N_1, N_2...N_n$ and $A^1, A^2...A^m$
OUTPUT: $N_1, N_2...N_n$ with RTS

for all A^i **do**
  $A^i_{max} = max\{A^i_1, A^i_2...A^i_n\}$,
  $A^i_{min} = min\{A^i_1, A^i_2...A^i_n\}$,
  $\mu(A^i) = \frac{\sum_{j=1}^{n} A^i_j}{n}$ and $\sigma(A^i) = \sqrt{\frac{\sum_{j=1}^{n} \left(A^i_j - \mu(A^i)\right)^2}{n}}$
end for
for all N_j **do**
  for all A^i **do**
    $Z(A^i_j) = \frac{A^i_j - \mu(A^i)}{\sigma(A^i)}$, $DS(A^i_j) = \frac{Z(A^i_j) - Z(A^i_{min})}{Z(A^i_{max}) - Z(A^i_{min})}$
  end for
  $RTS(N_j) = \frac{\sum_{i=1}^{m} DS(A^i_j)}{m}$
end for

Now the deciding attributes are selected as follows,

a) Attributes for WHO nodes: For WHO nodes we are assigning the trust in the range of $[0, 0.5]$. For docment type Videos we could not find any attributes. The attributes are,

Document type Boards and Microblogs

Twitter API[17] and Facebook API[16] are used to get some additional metadata of each Author from respective domains. The deciding attributes are as follows:

Number of Friends/Followers: It represents popularity of the author in the corresponding social network.

Follower to Following Ratio: Nobody relies on an author who is following 1000 accounts and has only 10 followers. So if the ratio of number of follower to the number of following is large then the author is more reliable.

Number of distinct tweets/posts: We calculate how many of the last 5 tweets or posts of the author are distinct. More is the number of distinct posts more reliable the Author is.

Document type Reviews

Generally review data come with *Author description* attribute which provides some information about the Author. For trip advisor data we get 5 deciding attributes:-

The *first field* represents the reputation of the reviewer i.e. whether he is a contributor of type **Top/Senior/Simple Contributor/Reviewer**. According to the presence of these 4 types of category $100, 70, 50$ and 30 are considered as numeric values respectively.

The *second, third* and *fourth* field represent the number of total reviews, the number of hotel reviews and the number of helpful votes posted by the reviewer so far.

The *fifth* field represents for how many distinct cities the reviewer has voted so far.

For other documents that do not belong to trip-advisor a default static value 0.2 is assigned as RTS.

b) Attributes for node type DOCUMENT: In this part we assign the trust in the range of $[0, 0.5]$. The deciding numeric attributes are as follows:-

Document type Boards, Reviews, Microblogs

Review Rating/Blog Rating: More the review/blog rating more reliable the document is.

Number of helpful votes: It represents how much useful the document is.

Post-In-Thread: It represents how much discussion has been taken place regarding the document. More the number of post in the thread more reliable the document is.

Similarity between title and actual text: It represents with how much relevance with its title the document is written.

Length of the content: Normally good documents are large as the author takes significant time to write them. Spams are more often small in size. So if the length of the content is large then probably the document is much reliable.

Number of paragraphs in content: Normally reliable documents are well structured and contain sections or paragraphs. But Spams usually contain many newline characters that is mistakenly interpreted as paragraph. So we also consider *Average length of paragraphs*.

Document type Videos

The attributes are *Proximity between Titlehtml and Texthtml* and *Comments-in-post*.

Attribute for node type WHERE

The only deciding attribute is *Pagerank score*. In this part we assign the trust in the range of $[0, 0.8]$.

Now on these selected attributes found so far we apply RTSA to find out RTS of WHO,DOCUMENT and WHERE node respectively. To make it in the required scale we multiply the calculated RTS value by $0.5, 0.5$ and 0.8 respectively. Now for every node we add the RTS and the MTS to calculate Trust Before Propagation (TBP).

	No. of total reviews	**No. of hotel reviews**	**No. of helpful votes**	**No. of distinct cities**
N1	4389	1289	125	64
N2	1792	852	145	78
N3	2428	219	81	66
N4	2243	192	180	37
N5	834	236	112	39
N6	1795	1529	95	12
N7	663	538	72	69
N8	3287	2462	113	39
N9	360	117	188	19
N10	955	271	123	21
mean	1874.6	770.5	123.4	44.4
sd	1267.66235	772.23359	38.48578	23.42933
max	4389	2462	188	78
min	360	117	72	12

3) Trust Propagation: TBP is propagated across different relationship edges depending on the relations and type of nodes associated. In this step, the trust is propagated in a range of $[0, 0.3]$. Let, N ={WHO,WHERE,DOCUMENT}, Z ⊆{USE, CISE, CONTENT_LOCATION_EDGE, PRE, DCSE, DRE}. The Propagated Trust Score Algorithm (PTSA) to calculate propagated trust score (PTS) between nodes of type N via edges Z is shown in Algorithm 3. Here, w is the scaling factor and $sim\_score(E)$ is the similarity value stored in the edge E.

Algorithm 3 Propagated Trust Score Agorithm (PTSA)

INPUT: RSGR with TBP
OUTPUT: RSGR with PTS
  for all $X \in N$ **do**
    if there exists any edge $E \in Z$ which connects node $Y \in N$ **then**
      Calculate number of such edges $N(E)$ associated with X and set the value of $sum = 0$
      for all E **do**
        Calculate $sum = sum + sim\_score(E) \times N(E)$
      end for
      Calculate $PTS = sum/N(E) \times w$
    else
      Set $PTS = 0$
    end if
  end for

The steps for finding the final trust are:-

a) From WHO node to WHO node: If two Authors are similar either in terms of interest or in terms of user-id and we assign some trust value to one author then some trust must be propagated to the other author depending on the strength of their relation. We apply PTSA to calculate PTS. Here, X, Y are WHO. Z is {USE, CISE} and $w = 0.3$.

b) From WHERE node to DOCUMENT node: The reliability of a document greatly depends on the domain where it gets published. It is pretty obvious that a *Wikipedia* document is much more reliable than a *chase.com* document even if they share the same content. We apply PTSA to calculate PTS. Here X is DOCUMENT and Y is WHERE, Z is CONTENT_LOCATION_EDGE, PRE, $w = 0.1$ and $sim\_score = 1$.

c) From DOCUMENT node to DOCUMENT: If two documents are similar either in terms of containing concepts and we assign some trust value to one document then some trust must be propagated to the other document depending on the strength of their relation. We apply PTSA to calculate the propagated trust. Here X and Y is DOCUMENT, Z is DCSE, DRE, $w = 0.1$ and $sim\_score(DRE) = 1$.

d) From WHO node to DOCUMENT node: There are two types of trust propagation. They are as follows,

Based on common concepts: If an Author is interested in a certain topic and he/she writes about it then we should rely the document. In other words some trust must be propagated from the author to the document. For each WHO node we check if the WHO node and the associated DOCUMENT node share any WHAT nodes. If they share then number of common WHAT nodes (say, $concept\_intersection\_count$) and the number of WHAT nodes associated with the DOCUMENT node (say $concept\_count$) are calculated. Then we propagate the trust of the WHO node multiplied by a static weight (0.05) and $(concept\_intersection\_count/concept\_count)$ to the associated DOCUMENT node. We then add the propagated trust to the existing trust of the DOCUMENT node.

Based on popularity of Author: If the Author is highly reliable i.e. the trust value of the WHO node is very high then whatever he/she writes we trust the document to some extent. In other words some trust has to be propagated from that WHO node to the associated DOCUMENT node. For each WHO node we check if the trust is very high (say greater than 0.8). Then we propagate the trust of the WHO node multiplied by a static weight (0.05) to the associated DOCUMENT node.

Finally we add the propagated trust with TBP of the DOCUMENT node and WHO node to calculate Final Trust.

IV. Experimental Results

In this section we show various experiments that support the effectiveness and the accuracy of our approach.

A. Experimental Setup

Data Sets. We use real life text documents especially from facebook.com, twitter.com, tripadvisor.com, amazon.com, youtube.com, dailymotion.com, hotel.com. It consists of $60,370$ unsupervised documents. After the feature extraction step the resulting graph database consists of $8,01,350$ nodes. For the purpose of experimental analysis we take a random sample of 600 documents consisting of 150 documents from each category i.e. Boards, Microblogs, Reviews and Videos. To test the accuracy of calculated trust, we adopt a user labeled dataset. In this dataset, those selected 600 documents are assigned 2 trust values one for the Author (i.e. WHO node) and other for the document itself (i.e. DOCUMENT node) by manually looking into the actual document and looking into the Authors profile. Assignment of trust values is done by 10 unbiased persons. The trust value assigned has range $[0, 1.0]$ and has precision up to one decimal place. These persons are not specialist in any domain and as the data used also do not belong to any specific category so we assume the labeling as the ground truth.

Method. At first we try to find out what amount of trust is assigned to corresponding WHO and DOCUMENT node for individual documents at every steps of our trust mining approach. Then we show the trust distribution of WHO and DOCUMENT node obtained at the end of each step of the discussed trust mining approach. Then we show the gradual improvement of the trust value assigned to the corresponding WHO and DOCUMENT node through the steps of our algorithm. Finally we show the accuracy of our algorithm.

Let for a particular document i and for a particular node type j, A_{ij} represents the trust value assigned by user and B_{ij} represents the trust value assigned by our algorithm. Then the accuracy is calculated as per Eq.6.

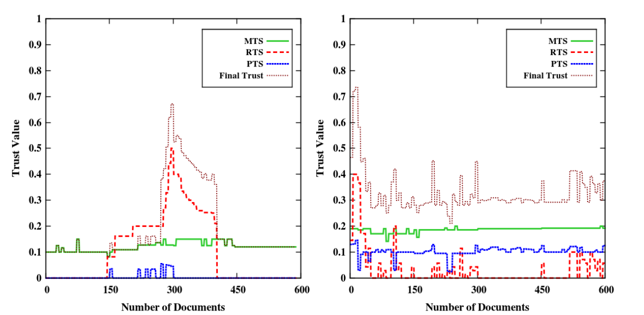
$$Accuracy_{ij} = 1 - \frac{|A_{ij} - B_{ij}|}{A_{ij}} \tag{6}$$

Let, N is the total number of nodes of type j. The overall accuracy for node type j is calculated as per Eq.7.

$$Overall\ Accuracy_j = \frac{\sum_{i=1}^{N} Accuracy_{ij}}{N} \tag{7}$$

B. Trust assigned at every step

The trust assigned only due to *MTSA, RTSA, PTSA* and the final trust are shown in Fig.4a and Fig.4b for WHO and DOCUMENT nodes respectively. The documents are sorted according to their respective types. Indices from (1-150) represent Boards, (151-300) represent Reviews, (301-450) represent Microblogs and (451-600) represent Videos.



(a) (b)

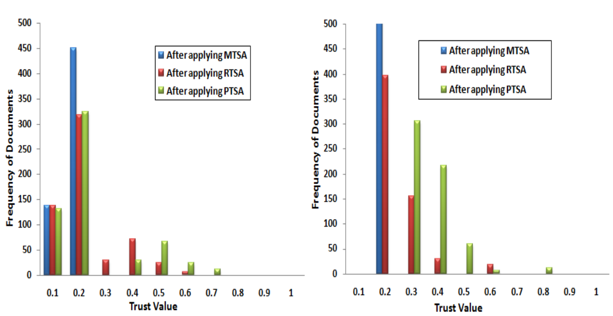
Figure 4: Trust assigned at every step of our algorithm for (a) WHO node; (b) DOCUMENT node

In Fig.4a for WHO node we get low MTS due to absence of some attributes.For Microblogs we have significant RTS

which are mainly twitter documents because we have mined information about the Author using twitter api. For some Review documents we get significant RTS which are mainly trip-advisor documents. For other Review documents we get default RTS as 0.2. Since for document types Boards and Videos we have much less information about the Author, we do not get any RTS in most of the cases. Still for some Boards documents we get some RTS which are mainly facebook documents as we have used facebook api. We have significant PTS via CISE in case of Reviews because for other type of documents *Author description* field is generally empty. For WHO node we have the most trustworthy Author (Trust value = 0.68) for document type Microblogs. In Fig.4b for DOCUMENT node we get significant MTS. We see that except for Microblogs all documents are assigned significant RTS. Microblogs are small documents so the number of paragraphs, number of external links etc are negligible compared to other documents. On the other hand for some of the document of type Boards, we have got RTS upto 0.4 as for those documents they are well written with paragraphs, having large number of comments etc. For most of the documents we get significant PTS. For some Boards and Reviews documents we get very low PTS because for those documents the corresponding WHERE node has very low Final Trust and they do not have much common concepts with the other documents. Here we have the most reliable document (Final Trust = 0.72) for document type Boards.

C. Trust distribution at the end of every step

The trust distribution of documents are shown in Fig.5a and Fig.5b for WHO and DOCUMENT nodes respectively. In Fig.5a for WHO node we see that after applying MTSA most of them have trust less than 0.2. After applying RTSA some of the WHO nodes are assigned more trust and few of them get trust value 0.5 and 0.6. After applying PTSA trust of some WHO nodes are further increased.



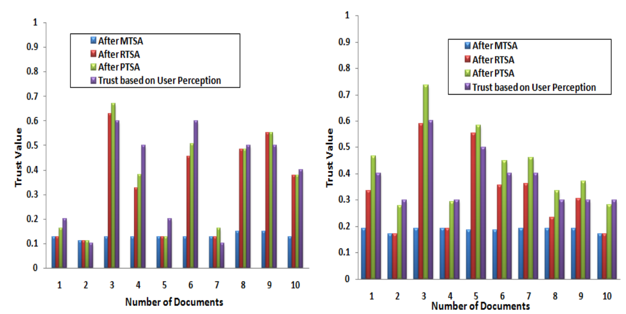
(a) (b)

Figure 5: Trust distribution for (a) WHO node; (b) DOCUMENT node

In Fig.5b for DOCUMENT node we see that after applying MTSA we have got trust almost equal to 0.2 for every document. After applying RTSA and PTSA we notice significant change in the trust distribution. If we are given individual documents in isolated manner we can find trust only based on the presence of metadata which would be very low. But if we are given all documents at the same time then we can calculate trust based on the richness of metadata as well as we can calculate trust of documents by comparing individual documents with the other documents. The low MTS for WHO and DOCUMENT and relatively improved trust after applying RTSA and PTSA supports this fact.

D. Improvement of Trust Value

We take a random sample of 10 documents to show the improvement of the trust value assigned to WHO and DOCUMENT node after end of each step of our algorithm as shown in Fig.6a and Fig.6b respectively. We compare these trust values with the trust based on user perception or actual trust.



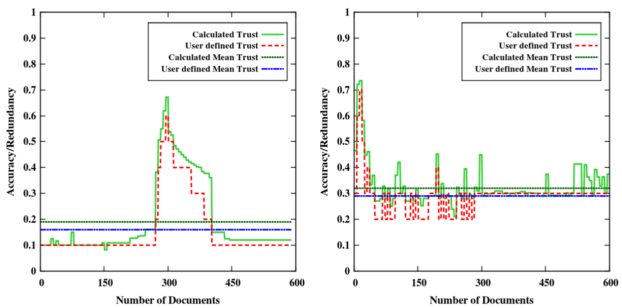
(a) (b)

Figure 6: Improvement of trust value for randomly selected 10 documents for (a) WHO node; (b) DOCUMENT node

In Fig.6a for WHO node and in Fig.6b for DOCUMENT node, MTS is very low compared to the actual trust in case of document number $3, 4, 6, 8, 9, 10$ and document number $1, 3, 5, 6, 7$ respectively. In Fig.6a for document number 4 and 6 we see even after trust propagation we do not get desirable trust. In Fig.6b for document number $1, 5, 6, 7, 8$ and 9 the trust after trust propagation exceeds the actual trust not with a large margin but in case of document number 3 we encounter some error. The possible reason for this is we are either increasing trust or leaving as it is. We are not decresing trust at any point because the social media documents are intrinsically not much reliable. The data we are dealing with are not much rich in terms of metadata. If we deal with some data which contain well structured attributes and highly rich metadata then we could implement the concept of decreasing trust. But overall we can clearly see that the difference between the user percepted trust and the calculated trust decreases after every step of our trust mining algorithm. This supports the effectiveness of our approach.

E. Accuracy

The comparison of the actual trust value with the trust value that we have calculated are shown in Fig.7a and Fig.7b for WHO and DOCUMENT nodes respectively. We see that the calculated trust values more or less follow the actual trust values.



(a) (b)

Figure 7: Comparison of calculated trust with the actual trust for (a) WHO node; (b) DOCUMENT node

In Fig.7a the *calculated mean trust value* is 0.19 and user defined or *actual mean trust value* is 0.16. In Fig.7b the *calculated mean trust value* is 0.32 and user defined or *actual mean trust value* is 0.29. The *overall accuracy* for WHO node is calculated to be 0.81 and for DOCUMENT node the *overall accuracy* is calculated to be 0.84 which is quite good. The possible reason for the decrease in accuracy could be we calculate trust value having continuous value and the user percepted trust value has precision of only single decimal place.

V. Conclusion

Finally we propose a three stage framework to mine trust from real life online documents step by step until we get trust distribution of Authors, Domains and the Content of the document itself. We generalized the concept of trust mining and developed an algorithm in which trust based on metadata, relative trust and propagated trust are considered to assign trust values. The experimental results show the effectiveness of our framework. We have used documents which are not very rich in terms of metadata overall. So we have used various api to mine additional information from respective domains. It is also the reason that we are not interested in penalizing trust. Our algorithm works fine even if there is large variation of the richness of metadata of the incoming documents.

Interesting problems related to the approach could be how to include the concept of geo-location proximity in quantifying trust. Another problem could be how to incorporate the concept of event detection in trust mining. There is a huge amount of unstructured data available in the online social media. In future, we shall use segmentation analysis, opinion mining and sentimental analysis to conduct more accurate trust mining. Due to availability of large amount of social media data we have very good platforms to carry out our further work.

Acknowledgment

This research is partially done in IBM's Extreme Blue Lab with Kiran Subbaraman. We thank the anonymous reviewer for labeling the documents.

References

[1] Golbeck J, Parsia B, Hendler J A. *Trust networks on the Semantic Web*, In Proc. the 7th Int. Workshop on Cooperative Intelligent Agents, Helsinki, Finland, 2003, pp.238-249.

[2] Guha R, Kumar R, Raghavan P, Tomkins A. *Propagation of trust and distrust*, In Proc. the 13th Int. Conference on World Wide Web, NY, USA, 2004, pp.403-412.

[3] Bizer C, Oldakowski R. *Using context- and content-based trust policies on the Semantic Web*, In Proc. the 13th Int.conference on World Wide Web, NY, USA, 2004, pp.228-229.

[4] Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G. *Finding high-quality content in social media*, Proc. of the Intl. Conf. on Web Search and Web Mining, pp. 183194. (2008)

[5] Blumenstock, J. *Size matters, word count as a measure of quality on Wikipedia*, Proc. of the 17th Intl. Conf. on World Wide Web, pp. 10951096. ACM, New York, 2008.

[6] Zolfaghar K, Aghaie A. *Mining trust and distrust relationships in social web applications,* In Proc. the 2010 IEEE Int. Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, Aug. 26-28, 2010, pp.73-80.

[7] McGuinness, D., Zeng, H., Da Silva, P., Ding, L., Narayanan, D., Bhaowal, M. *Investigations into trust for collaborative information repositories: a Wikipedia case study,* In: Proc. of the Workshop on Models of Trust for the Web 2006, pp. 3-131.

[8] Nguyen V A, Lim E P, Jiang J, Sun A. *To trust or not to trust? Predicting online trusts using trust antecedent framework,* In Proc. the 9th IEEE Int. Conference on Data Mining, Miami, Florida, USA, Dec. 6-9, 2009, pp.896-901.

[9] Chua F C T, Lim E P. *Trust network inference for online rating data using generative models*, In Proc. The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010, pp.889-898.

[10] Matsuo Y, Yamamoto H. *Community gravity: Measuring bidirectional effects by trust and rating on online social networks*, In Proc. the 18th Int. Conference on World Wide Web, Madrid, Spain, April 20-24, 2009, pp.751-760.

[11] Zhang Y, Yu T. *Mining trust relationships from online social networks*, JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(3): 492-505 May 2012.

[12] Wu, Z., Palmer, M. *Verb semantics and lexical selection*, In: Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics. (1994) pp.133-138.

[13] Sai T. Moturu, Huan Liu, *Evaluating the Trustworthiness of Wikipedia Articles through Quality and Credibility*, In Proceedings of the 5th International Symposium on Wikis and Open Collaboration, NY, 2009.

[14] Sharma T, Toshniwal D. *A Generalized Relationship Mining Approach for Social Media Text Documents*, In Proc. 3rd International Congress on Big Data, June 27 - July 2, 2014, Anchorage, Alaska, USA. (unpublished).

[15] http://opennlp.apache.org/

[16] https://developers.facebook.com/docs/reference/apis/

[17] https://dev.twitter.com/docs/streaming-apis

[18] https://www.quantcast.com/top-sites