2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

A Visual Framework for Clustering Memes in Social Media

Anh Dang[1], Abidalrahman Moh'd[1], Anatoliy Gruzd[2], Evangelos Milios[1], and Rosane Minghim[3]

[1]{anh,amohd,eem}@cs.dal.ca, Dalhousie University, 6050 University Avenue, Halifax, NS, B3H 4R2, Canada
[2]gruzd@ryerson.ca, Ryerson University, 55 Dundas St. West, Toronto, ON, M5G 2C5, Canada
[3]rminghim@icmc.usp.br, University of São Paulo-USP, ICMC, São Carlos, Brazil

***Abstract*—The spread of "rumours" in Online Social Networks (OSNs) has grown at an alarming rate. Consequently, there is an increasing need to improve understanding of the social and technological processes behind this trend. The first step in detecting rumours is to identify and extract memes, a unit of information that can be spread from person to person in OSNs. This paper proposes four similarity scores and two novel strategies to combine those similarity scores for detecting the spread of memes in OSNs, with the end goal of helping researchers as well as members of various OSNs to study the phenomenon. The two proposed strategies include: (1) automatically computing the similarity score weighting factors for four elements of a submission and (2) allowing users to engage in the clustering process and filter out outlier submissions, modify submission class labels, or assign different similarity score weight factors for various elements of a submission using a visualization prototype. To validate our approach, we collect submissions on Reddit about five controversial topics and demonstrate that the proposed strategies outperform the baseline.**

I. Introduction

Online social networks (OSNs) are networks of online interactions and relationships that are formed and maintained through various social networking sites such as Facebook, LinkedIn, Reddit, and Twitter. Nowadays, hundreds of millions of people and organizations turn to OSNs to interact with one another, share information, and connect with friends and strangers. OSNs have been especially useful for disseminating information in the context of political campaigning, news reporting, marketing, and entertainment.

Despite their popularity, OSNs have also a "negative" side. As well as spreading credible information, OSNs can also spread rumours, which are truth-unverifiable statements. For example, so many rumour-driven memes about swine flu outbreak (e.g., "swine flu pandemic meme") were communicated via OSNs in 2009 that the US government had to tackle it officially on their website [1], [2]. Problems like these (i.e., rumours going viral) are unfortunately not isolated and prompt the question of how to identify and limit the spread of rumours in OSNs. In order to detect rumours, we have to identify memes that are rumour-related in OSNs. Clustering is a simple and efficient unsupervised process to identify memes in OSNs by grouping similar information into the same category. However, traditional clustering algorithms do not work effectively in OSNs due to the heterogeneous nature of social network data [3]. Labelling massive amounts of social network data is an intensive task for classification. To overcome these limitations, this paper proposes a semi-supervised approach with relevance user feedback for detecting the spread of memes in OSNs.

In text clustering, a similarity measure is a function that assigns a score to a pair of texts in a corpus that shows how similar the two texts are. Computing similarity scores between texts is one of the most computationally intensive and important steps for producing a good clustering result. For a meme clustering task, this process is usually hindered by the lack of significant amounts of textual content, which is an intrinsic characteristic of OSNs [3]. For example, in Reddit[1], most submission titles are very short and concise. Although the title of a submission may provide meaningful information about the topic, titles may not provide enough information to determine if two submissions are discussing the same topic. The sparsity of Reddit submission title texts significantly contributes to the poor performance of traditional text clustering techniques for grouping submissions into the same category. We therefore propose a strategy to leverage the use of references to external content.

A submission may include one or more comments from users that discuss the submission topic. It can also contain a URL that points to an external article that further discusses the topic of the submission. Similarly, a submission may include an image that also provides more valuable information about the submission topic. By introducing the use of comments, URL content, and image content of a submission, we exploit more valuable data for text clustering tasks, which helps detect memes in OSNs more efficiently.

Vector space models are commonly used to represent texts for a clustering task. In these models, each text is represented as a vector where each element corresponds to an attribute extracted from the text. One of the benefits of these models is their simplicity in calculating the similarity between two vectors based on linear algebra. The two most famous models are term frequency - inverse term frequency (TF-IDF) and Bag-of-Words. However, those models rely solely on lexical representation of the texts, which does not capture the importance of the semantic relatedness between words in a text. For example, the use of polysemy and synonymy are very popular in several types of texts and play an important role in determining whether two words, concepts or texts are semantically similar. This motivates many researchers to explore the advantage of semantic similarity in the task of

[1]Reddit - http://www.reddit.com

ASONAM '15, August 25-28, 2015, Paris, France
 ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2808830

text clustering by utilizing word relatedness through external thesauruses like WordNet and Wikipedia [4], [5]. However, they remain far from covering every word and concept used in OSNs. Islam et al. [6] used Google n-grams dataset to compute the relatedness between words. This paper explores Google n-grams algorithm of Islam et al. for computing similarity scores and proposes two novel strategies to combine those scores for the task of clustering memes.

With the increasing amount of online social network data, understanding and analyzing them are becoming more challenging. Researchers have started to employ human's ability to effectively gain visual insight on data analysis tasks. In the case of spreading memes in OSNs, related memes are very similar. Visualizing the relationships between texts provides a better understanding of the data. The task of clustering memes shares some similarity with clustering texts, but they are also intrinsically different. For example, social network data is usually poorly-written and content-limited. This reduces the quality of the clustering results. For a Reddit submission, the relationships between the title, comment, image, and URL sometimes are disconnected (e.g., a title has a different meaning from the URL). In this paper, we developed a visualization prototype[2] to allow users to better distinguish the similarity between submissions and use this feedback to improve the clustering results.

This paper formalizes the problem of meme clustering and proposes a novel approach for clustering Reddit submissions. It makes the following contributions:

- Automatically annotates a meme dataset about five controversial topics in Reddit.
- Defines several similarity scores between submissions, leveraging various types of external content for meme clustering tasks.
- Proposes Internal Centrality-Based Weighting and Similarity Score Reweighting with Relevance User Feedback strategies which combine similarity scores between submissions to better represent the submission semantic content.
- Compares the proposed similarity scores between submissions using Google Tri-gram Method [6] against Euclidean similarity for TF-IDF vectors.

II. Related Work

This section presents current research on semantic text similarity and detecting the spread of memes in OSNs.

A. Similarity measures and Text Clustering

Several similarity measures have been proposed in the literature for the task of text clustering. The most popular ones are lexical measures like Euclidean, Cosine, Pearson Correlation, and Extended Jaccard measures. Strehl et al. [7] provided a comprehensive study on using different clustering algorithms with these four similarity measures. The authors used several clustering algorithms on the YAHOO dataset, and showed that Extended Jaccard and Cosine similarity performed better and achieved results that are close to a human-labelling process. However, lexical similarity measures do not consider the semantic similarity between words in the texts.

Some researchers have taken advantage of the semantic relatedness of texts by using external resources to enrich word representation. In [4], the authors suggested using WordNet as a knowledge base to determine the semantic similarity between words. The experiment results have shown that an external knowledge base like WordNet improves the clustering results in comparison to the Bag-of-Words models. Hu et al. [5] proposed the use of Wikipedia as an external knowledge for text clustering. The authors tried to match concepts in texts into Wikipedia concepts and categories. The similarity scores between concepts are calculated based on the text content information, Wikipedia concepts, and categories. The experiment results have shown that using Wikipedia as an external knowledge provided a better result than using WordNet due to the limited coverage of WordNet. Bollegala et al. [8] proposed the use of information available on the Web to compute semantic similarity between texts by exploiting the page counts and text snippets returned by a search engine. Our work is intuitively different from these approaches, as it introduces the use of word relatedness based on the Google n-grams dataset [9]. The proposed semantic similarity scores between texts are calculated based on that algorithm to handle the low quality (i.e. poor writing) of social network data. We argue that it is more effective than textual as well as other semantic approaches, as the Google n-grams dataset has more coverage than other semantic approaches.

B. Detecting memes in Online Social Networks

The amount of rumour-driven communication delivered through OSNs has increased considerably lately. The first step in detecting a rumour is to identify the emerging memes in OSNs. Cataldi et al. [10] proposed an approach that monitored the real-time spread of emerging memes in Twitter. The authors defined an emerging term as one whose frequency of appearance had risen within a short period of time and had not emerged or was only rarely discussed in the past. A navigable topic graph is constructed to connect semantically related emerging terms. Emerging memes are extracted from this graph based on semantic relationships between terms over a specified time interval. Leskovec et al. [11] proposed a meme-tracking framework to monitor memes that travel through the Web in real-time. The framework studied the signature path and topic of each meme by grouping similar short distinctive phrases together. One drawback of this framework is that it only applies lexical content similarity to detect memes. This did not work well for memes that are related but not using the same words, and those that are short and concise (e.g., Tweets on Twitter).

To address this limitation, JafariAsbagh et al. [3] introduced the concept of Protomeme to tackle the sparsity of text in Twitter. Each Protomeme is defined based on one of the atomic information entities in Twitter: hashtags, mentions, URLs, and tweet content. An example of Protomeme is the set of tweets containing the hashtag #All4Given. They also proposed several similarity measures and their combinations, based on Protomeme to group semantically and structurally related tweets. Our proposed approach shares some commonalities

[2]The visualization prototype - https://youtu.be/ej7LlPOPikI

$$\mathrm{GTM}(\omega_1,\omega_2)=\begin{cases}\dfrac{\log\frac{\mu_T(\omega_1,\omega_2)C_{\max}^2}{C(\omega_1)C(\omega_2)\min(C(\omega_1)C(\omega_2))}}{-2\times\log\frac{\min(C(\omega_1),C(\omega_2))}{C_{\max}}} & \text{if } \log\frac{\mu_T(\omega_1,\omega_2)C_{\max}^2}{C(\omega_1)C(\omega_2)\min(C(\omega_1)C(\omega_2))}>1\\ \dfrac{\log 1.01}{-2\times\log\frac{\min(C(\omega_1),C(\omega_2))}{C_{\max}}} & \text{if } \log\frac{\mu_T(\omega_1,\omega_2)C_{\max}^2}{C(\omega_1)C(\omega_2)\min(C(\omega_1)C(\omega_2))}\leq 1\\ 0 & \text{if } \mu_T(\omega_1,\omega_2)=0\end{cases}$$

Fig. 1: GTM semantic similarity calculation [6].

with their work in that it adopts the use of semantic similarity measures between submissions on Reddit to detect memes in OSNs. Although Twitter has been the most popular OSN for detecting memes, little work has been done to detect rumour-related memes on Reddit.

Researchers also explored the use of visualization for clustering texts with relevance user feedback. Lee et al. [12] introduced iVisClustering, an interactive visualization framework based on LDA topic modelling. This system provides some interactive features, such as removing a useless document or a cluster, moving a document from one cluster to another, and merging two clusters. Choo et al. [13] presented an interactive visualization for dimension reduction and clustering for large-scale high-dimensional data. The system allows users to interactively try different dimension reduction techniques and clustering algorithms to optimize the clustering results. One of the limitations of those systems is that they focus on the clustering algorithms and results and have limited supports for combining similarity scores for different parts of a text (e.g., the title and body of a text). This paper introduces a visualization prototype to combine different similarity scores for our clustering process interactively and incrementally.

III. Reddit Social Network

Reddit, which claims to be "the front page of the internet", is a social news website, where users, called redditors, can create a submission or post direct links to other online content. Other redditors can comment or vote to decide the rank of this submission on the site. Reddit has many subcategories, called sub-reddits that are organized by areas of interests. The site has a large base of users who discuss a wide range of topics daily, such as politics and world events. Alexa ranks Reddit as the 24$^{\text{th}}$ most visited site globally[3]. Each Reddit submission has the following elements:

- **Title:** The title summarizes the topic of that submission. The title text is usually very short and concise. Title may also have a description to further explain it.
- **Comments:** Users can post a comment that expresses their opinions about the corresponding submission or other user comments. Users can also vote comments up or down.
- **URL:** Each submission may contain a link to an external source of information (e.g., news articles) that is related to the submission.
- **Image:** Submissions may also have a link to an image that illustrates the topic of the submission.

Unlike other OSNs, Reddit is fundamentally different in that it implements an open data policy; users can query any posted data on the website. For example, other OSNs, like Twitter or Facebook, allows circulating information through a known cycle (e.g., "friend" connections), whereas Reddit promotes a stream of links to all users in a simple bookmarking interface. This makes Reddit a more effective resource to study the spread of memes in OSNs. To the best of our knowledge, no similar work has been done on clustering memes in Reddit.

[3]Reddit ranking - http://www.alexa.com/siteinfo/reddit.com

IV. Meme Detection Framework

The meme detection problem is defined for any social media platform used to spread information. In these systems, users can post a discussion or discuss a current submission. Fig. 2 shows an overview of the proposed meme detection framework.

A. Google Tri-gram Method

Google Tri-gram method (GTM) is an unsupervised corpus-based approach for computing semantic relatedness between texts. GTM uses the uni-grams and tri-grams of the Google Web 1T N-grams corpus [6] to calculate the relatedness between words, and then extends that to longer texts. The Google Web 1T N-grams corpus contains the frequency count of English word n-grams (unigrams to 5-grams) computed over one trillion words from web page texts collected by Google in 2006.

The relatedness between two words is computed by considering the tri-grams that start and end with the given pair of words, normalizing their mean frequency with unigram the frequency of each of the words as well as the most frequent unigram in the corpus as shown in Fig. 1, where $C(\omega)$ is the frequency of the word ω. $\mu_T(\omega_1,\omega_2)$ is the mean frequency of trigrams that either start with ω_1 and end with ω_2, or start with ω_2 and end with ω_1. $\sigma(a_1,\ldots,a_n)$ is the standard deviation of numbers $a_1,\ldots,a_n$, and $C_{\max}$ is the maximum frequency among all unigrams. GTM computes a score between 0 and 1 to indicate the relatedness between two texts based on the relatedness of their word content. For given texts P and R where $|P|\leq|R|$, first all the matching words are removed, and then a matrix with the remaining words $P'=\{p_1,p_2,\cdots,p_m\}$ and $R'=\{r_1,r_2,\cdots,r_n\}$ is constructed where each entry is a

GTM word relatedness $a_{ij} \leftarrow GTM(p_i, r_j)$.

$$M = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

From each row $M_i = \{a_{i1} \cdots a_{in}\}$ in the matrix, significant elements are selected if their similarity is higher than the mean and standard deviation of words in that row:

$$A_i = \{a_{ij} | a_{ij} > \mu(M_i) + \sigma(M_i)\},$$

Where $\mu(M_i)$ and $\sigma(M_i)$ are the mean and standard deviation of row i. Then the document relatedness can be computed using:

$$Rel(P,R) = \frac{(\delta + \sum^m a_{i=1}\sigma(A_i)) \times (m+n)}{2mn}$$

Where $\sum^m a_{i=1}\sigma(A_i)$ is the sum of the means of all the rows, and δ is the number of removed words when generating P' or R'.

B. Similarity Scores

This section explores the use of GTM semantic similarity to propose four semantic similarity scores and their combinations between submissions. Representing a submission S in Reddit as a vector $S = (T, C, I, U)$ where:

- T is an n-dimensional feature vector $t_1, t_2, ...t_n$ representing the title of the submission and its description.
- C is an n-dimensional feature vector $c_1, c_2, ...c_n$ representing the comments of a submission.
- U is an optional n-dimensional feature vector $u_1, u_n, ...u_n$ representing the external URL content of a submission.
- I is an optional n-dimensional feature vector $i_1, i_2, ...i_n$ representing the image content of a submission. This content is extracted by using Google Reverse Image Search, which takes an image as a query and extracts the text content of the website that is returned from the top search result and is not from Reddit.

Next, we propose four similarity measures between two submissions S_1 and S_2:

- **Title similarity** S_t is the GTM semantic similarity score between the title word vectors T_1 and T_2.
- **Comment similarity** S_c is the GTM semantic similarity score between the comment word vectors C_1 and C_2.
- **URL similarity** S_u is the GTM semantic similarity score between the URL content word vectors U_1 and U_2.
- **Image similarity** S_i is the GTM semantic similarity score between the word vectors I_1 and I_2 retrieved from Google Reverse Image Search.

C. Similarity Scores and Combination Strategies

The main goal of this section is to study the effect of different similarity scores and their combinations on the quality of the meme clustering tasks.

1) Pairwise Maximization Strategy: The pairwise maximization strategy chooses the highest among the title, comment, URL, and image scores to decide the similarity between two submissions. This strategy avoids the situation where similarity scores have a low content quality (e.g., titles are short and lack details, comments are noisy, images and URLs are not always available) by choosing the most similar among them.

Given two submissions $S_1 = \{T_1, C_1, I_1, U_1\}$ and $S_2 = \{T_2, C_2, I_2, U_2\}$, the pairwise maximization strategy between them is defined as:

$$MAX_{S_1 S_2} = MAX(GTM_{T_1T_2}, GTM_{C_1C_2}, GTM_{U_1U_2}, GTM_{I_1I_2}) \quad (1)$$

Where $GTM_{T_1T_2}, GTM_{C_1C_2}, GTM_{U_1U_2}, GTM_{I_1I_2}$ are the title, comment, URL, and image similarity scores between the two submissions S_1 and S_2.

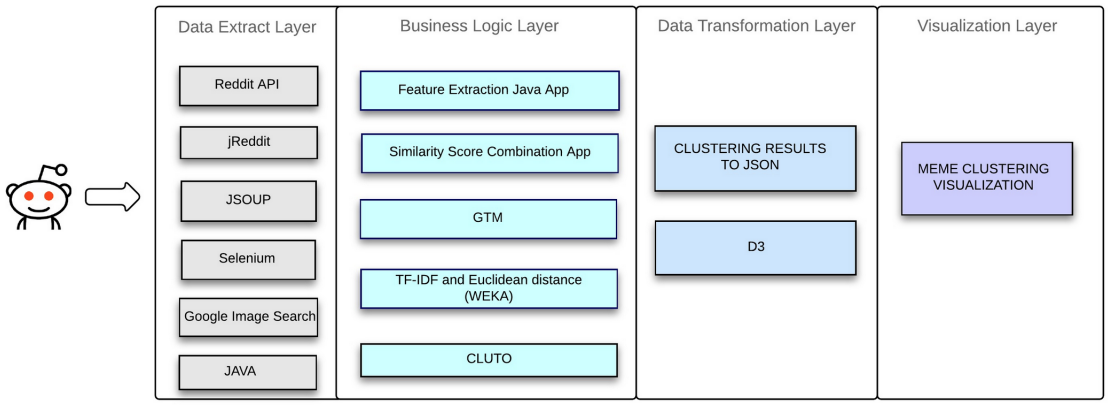


Fig. 2: The proposed meme detection framework.

2) Pairwise Average Strategy: The pairwise average strategy computes the average value of the four pairwise similarity scores. This strategy balances the scores among the four similarities in case some scores do not reflect the true content of the submission. It is defined as follows:

$$AVG_{S_1 S_2} = AVG(GTM_{T_1T_2}, GTM_{C_1C_2}, GTM_{U_1U_2}, GTM_{I_1I_2}) \quad (2)$$

3) Linear Combination Strategy: In the linear combination strategy, users can assign different weighting values manually. For example, if users think the title text does not capture the topic of a submission, they can assign a low weight factor (e.g., 0.1). If they think comment texts are longer and represent the topic better, they can assign a higher weight factor (e.g., 0.6). The linear combination strategy is defined as follows:

$$LINEAR_{S_1 S_2} = LINEAR(w_tGTM_{T_1T_2}, w_cGTM_{C_1C_2}, w_uGTM_{U_1U_2}, w_iGTM_{I_1I_2}) \quad (3)$$

Where w_t, w_c, w_u, and w_i are the weighting factors for S_t, S_c, S_u, and S_i with a normalization constraint $w_\mathrm{t}+w_\mathrm{c}+w_\mathrm{u}+w_\mathrm{i}=1$.

4) Internal Centrality-Based Weighting: Computing the optimized weight factors for the linear combination strategy is an intensive task. JafariAsbagh et al. [3] used a greedy optimization algorithm to compute the optimized linear combination for the task of clustering memes. However, it is unrealistic to compute all the possible weighting combinations for Equation 3. To alleviate this computational cost, we propose the Internal Centrality-Based Weighting (ICW), a novel strategy to automatically calculate the weight factors of the linear combination strategy. This strategy calculates the weight factors for each element of a submission by considering its surrounding context. Although all elements of a submission are semantically related, some elements could have more semantic content than others; for example, the URL content discusses more about the topic than the title. More weight is assigned to the elements with higher semantic content. Equation 4 shows the proposed strategy. It computes the semantic content weights using internal and external similarity scores between titles, comments, URLs, and images of two submissions. For each submission, this strategy computes the centrality score for each element of each submission S_i :

$$CENT_{\mathrm{T_i}} = GTM_{\mathrm{TC_i}} + GTM_{\mathrm{TU_i}} + GTM_{\mathrm{TI_i}}$$

$$CENT_{\mathrm{C_i}} = GTM_{\mathrm{CT_i}} + GTM_{\mathrm{CU_i}} + GTM_{\mathrm{CI_i}}$$

$$CENT_{\mathrm{U_i}} = GTM_{\mathrm{UT_i}} + GTM_{\mathrm{UC_i}} + GTM_{\mathrm{UI_i}}$$

$$CENT_{\mathrm{I_i}} = GTM_{\mathrm{IT_i}} + GTM_{\mathrm{IC_i}} + GTM_{\mathrm{IU_i}}$$

Then, it computes the weighting factors between two submissions S_1 and S_2 by:

$$w_\mathrm{T} = CENT_{\mathrm{T_1}} * CENT_{\mathrm{T_2}}$$

$$w_\mathrm{C} = CENT_{\mathrm{C_1}} * CENT_{\mathrm{C_2}}$$

$$w_\mathrm{U} = CENT_{\mathrm{U_1}} * CENT_{\mathrm{U_2}}$$

$$w_\mathrm{I} = CENT_{\mathrm{I_1}} * CENT_{\mathrm{I_2}}$$

Then, it normalizes the weighting factors so that: $w_\mathrm{T}+w_\mathrm{C}+w_\mathrm{U}+w_\mathrm{I}=1$, and finally computes the ICW strategy:

$$ICW_{S_1 S_2} = ICW(w_TGTM_{T_1T_2}, w_CGTM_{C_1C_2}, w_UGTM_{U_1U_2}, w_IGTM_{I_1I_2}) \quad (4)$$

5) Similarity Score Reweighting with Relevance User Feedback: One effective way to improve the clustering results is to manually specify the relationships between pairwise documents (e.g., must-link and cannot-link) to guide the document clustering process [14]. As social network data are intrinsically heterogeneous and multidimensional, it is not easy to compare two submissions to determine if they are similar or not without putting them into the same context. To overcome this limitation, a novel technique, the Similarity Score Reweighting with Relevance User Feedback (SSR), is proposed to incorporate relevance user feedback by a visualization prototype in which submissions are displayed as a force-directed layout graph where:

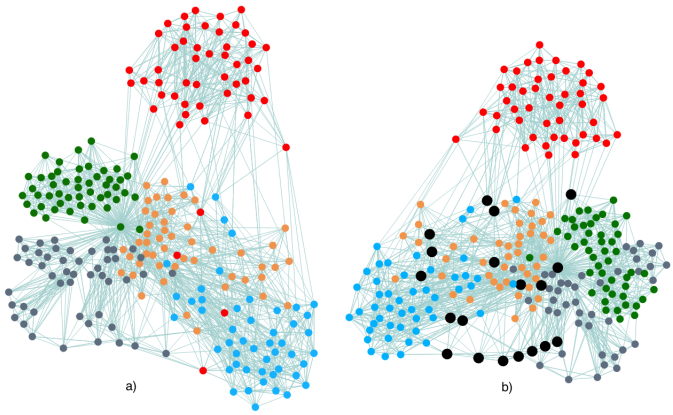
- **A node** is a submission in Reddit.
- **An edge** is a connection between two submissions if their similarity scores are above a threshold (default 0.85).
- **A node colour** represents to which cluster it belongs.

Algorithm 1 describes how the visualization system integrates user feedback to remove outliers, move submissions from a cluster to another, or reassign similarity score weighting factors for submissions. Users can select any of the four proposed strategies, MAX, AVG, LINEAR, and ICW as a baseline for clustering. Fig. 3 (a) shows an SSR visualization of the meme dataset using ICW strategy. The graph has five different colours that represent five memes in the ground-truth dataset. Users can pan, zoom, or click on a submission to get more details about this submission. They can also click on the checkbox "Show wrong cluster assignments" to see which submissions are incorrectly assigned by the ICW strategy. Based on the graph visualization, users can understand how a submission is positioned regarding its neighbour submissions. When clicking on a node in the graph, users will be redirected to the actual submission in Reddit to find out more information and decide if it belongs to the correct cluster. Most of the incorrectly clustered are overlapped or outlier nodes as shown in Fig. 3 (b). For each incorrectly assigned submission, users can remove, update

Algorithm 1 Semi-supervised Similarity Score Reweighting with Relevance User Feedback strategy (SSR).

Input: a set of submissions X from Reddit.
Output: K clusters $\{\mathrm{X}\}_{l=1}^{K}$

1: **loop**
2: {**Step 1**} Perform hierarchical clustering on C percent of the ground-truth dataset using one of the proposed strategies. C is defined through experiments.
3: {**Step 2**} Visualize the clustering result in step 1.
4: {**Step 3**} Allow users to interactively remove outlier submissions, reassign submission class labels, or assign weight factors for each element between two submissions.
5: {**Step 4**} Re-cluster the submissions based on user inputs.
6: {**Step 5**} repeat step 1 if necessary.
7: **end loop**
8: {**Step 6**} Recluster the whole dataset considering user feedback in Step 1 to 5.



(5)

Fig. 3: The proposed meme visualization: a) The original visualization graph and b) The clustering result using ICW

its class label, or assign a different similarity coefficient score for each element between two submissions. SSR focuses on human knowledge to detect outliers or borderline submissions.

6) Clustering Algorithms: The paper adopts hierarchical clustering as the baseline-clustering algorithm because of its simplicity and low computational cost. The experiments focused more on determining if the proposed similarity strategies improve clustering results. GTM is used to compute similarity between texts. The output of the GTM algorithm is a similarity matrix that shows the similarity score for each text with the other texts in the dataset. After producing this similarity matrix, gCLUTO[4] is used to cluster the matrix using hierarchical clustering.

V. Experimental Results

The objective of this section is to evaluate the performance of the meme clustering tasks. First, we explain how the ground-truth dataset is extracted from Reddit, and then discuss the evaluation metric and how the experiments are carried out.

A. Ground-truth dataset

To study the spread of memes in Reddit, the posts and comments related to a specific meme are identified. A generic query is used to capture all of the related submissions for a specific meme. Since there are no available Reddit meme datasets, Reddit API[5] and jReddit[6], an open source java project, are used to extract submissions, comments, and other data views (image and URL content) about a specific meme using predefined regular expressions. All submissions that do not have any comments or "up" or "down" votes are removed, as we assume that users are not interested in them. In addition, comments less than 5 words long are ignored. Stop words are also removed. For each submission with a URL in its title, JSOUP[7] is used to parse the main body text content of the URL. Occasionally, a submission can have an image in its title. Selenium[8] is used to submit the image to Google Reverse Image Search to find the most similar webpage to this image. If the top-searched result returns an article from Reddit, the program traverses through the search result list until it finds an article that is not from Reddit.

The ultimate goal of this framework is to detect memes and discussion topics online. In order to access the performance of the proposed similarity strategies, we collect ground-truth data for the experiments. First, the five most popular topics in Reddit from October to November 2014 are selected. The program extracts titles, comments, URL, and image content of all related submissions for each topic. Each topic is labeled to the corresponding cluster based on the keyword search. The five topics (clusters) are: (1) EBOLA (2) Ferguson (3) ISIS (4) Obama and (5) Trayvon Martin. Table I shows the detailed statistics of the ground-truth dataset.

B. Baseline

For the baseline, each title, comment, URL, or image text is represented as a TF-IDF vector. Euclidean distance [15] is used to calculate the similarity score between TF-IDF vectors due to its simplicity. In the next section, we compare the clustering results using GTM score and the baseline.

C. Results

For this ground-truth dataset, since the class labels exist for all of the submissions, purity is adopted (i.e., the number of correctly assigned submissions over the total number of submissions) as an evaluation measure. Larger purity value indicates better clustering results. Several configurations are explored to evaluate the performance of the proposed similarity strategies. As URL and image content are not always available, they are used as additional data for the clustering tasks for MAX, AVG, and ICW. The proposed configurations are used for both GTM similarity and baseline similarity and configured as:

- **TITLE**: Only use the title similarity for pairwise submission comparison.

[4]gCLUTO - http://glaros.dtc.umn.edu/gkhome/cluto/gcluto/overview
[5]RedditAPI - https://www.reddit.com/dev/api
[6]jReddit - https://github.com/karan/jReddit
[7]JSOUP - http://jsoup.org/
[8]Selenium - http://www.seleniumhq.org/

TABLE I: The experiment ground-truth dataset.

No.	Topic	Submission Counts	Comments	Submissions with a URL	Submissions with an image
1	EBOLA	495	89394	218	39
2	FERGUSON	495	83912	203	87
3	ISIS	488	76375	190	61
4	OBAMA	490	139478	142	13
5	Trayvon Martin	471	93848	250	30

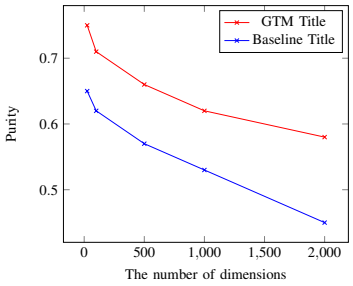


Fig. 4: GTM Title vs. Baseline Title

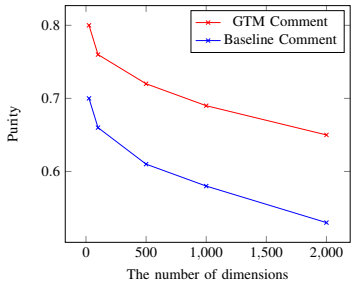


Fig. 5: GTM Comment vs. Baseline Comment

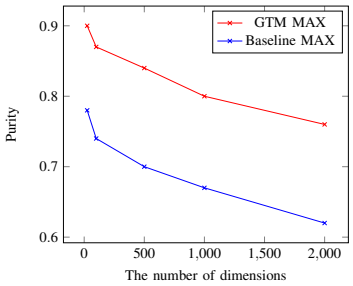


Fig. 6: GTM MAX vs. Baseline MAX

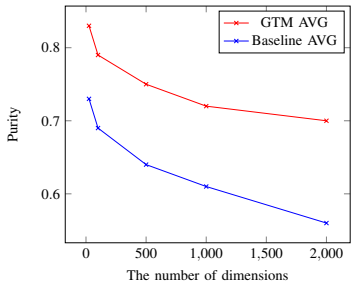


Fig. 7: GTM AVG vs. Baseline AVG

- **COMMENT**: Only use the comment similarity for pairwise submission comparison.
- **MAX**: Use the maximum of the four similarity scores for pairwise submission comparison as defined in Equation 1.
- **AVG**: Use the average of the four similarity scores for pairwise submission comparison as defined in Equation 2.
- **ICW**: Calculate the pairwise similarity between two submissions based on internal centrality weighting as defined in Equation 4.

1) Clustering with semantic similarity scores: The first experiment explores the advantage of using GTM semantic similarity for meme clustering tasks. The hierarchical clustering results between the GTM semantic similarity scores and the baselines using TF-IDF and Euclidean are compared for TITLE, COMMENT, MAX, and AVG. GTM semantic similarity score outperforms the baselines as shown in Fig. 4 and 5. Using comment content for clustering produces a better result than using title content as title texts are usually concise and does not represent the context of a submission. Exploiting additional image and URL content by AVG and MAX strategies improves the clustering results as shown in Fig. 6 and 7. Another interesting result is that the performance of GTM for comments is very close to the GTM AVG strategy. Using AVG strategy does not capture the semantic content of each similarity score efficiently. The experiment results also show that GTM semantic similarity score scales better than the baseline for higher vector dimensions. We conjecture that GTM helps alleviate "the curse of dimensionality" for clustering using traditional similarity measures.

2) The proposed ICW Strategy: In this experiment, the objective is to find out if the proposed ICW strategy improves the clustering result for a meme clustering task. The experiment results between the proposed ICW, AVG, and MAX are shown in Fig. 8. Results indicated that ICW outperforms AVG and achieves better results than MAX. We also found that the AVG combination does not provide good results when comparing with using MAX or ICW. One of the reasons may be each similarity score plays a different role in distinguishing memes on Reddit and this agrees with our assumption about the semantic content related between elements of submissions.

3) Similarity Score Reweighting with Relevance User Feedback: This section investigates the improvement from using user feedback with the visualization prototype for the meme clustering task. At first, users can select one of the four proposed strategies (AVG, MAX, LINEAR and ICW) to cluster the ground-truth dataset. For the LINEAR, we explore different weight factors for the Equation 3. Although the clustering results are improved when weight factors for title and comment

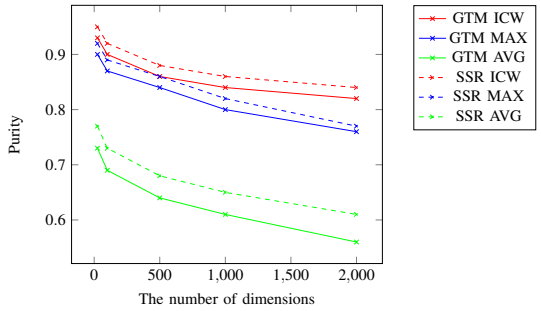


Fig. 8: Clustering results with different similarity score strategies.

are low (e.g., 0.1 for titles, 0.3 for comments) and are high for URLs and image (e.g., 0.6), their results are still not optimized when comparing with MAX and ICW. We remove outliers and reassign the weight factors for overlapping nodes. The clustering results are statistically significantly improved for both MAX, AVG, and ICW at p = 0.05 as shown in Fig. 8 (SSR ICW, SSR MAX, SSR AVG).

VI. Conclusion

This paper presents a framework to tackle the problem of meme clustering in Reddit as a means to detect rumours and their spread as specific clusters. This framework introduces the use of Google Web 1T N-grams corpus to compute the similarity between texts for the meme clustering task. It also defines several pairwise similarity scores between elements of two submissions. These scores could include external content related to image and URL elements of a submission. The paper explores several strategies to combine the similarity scores in order to produce better clustering results. These strategies include average, maximum, linear combination, internal centrality-based weighting, and similarity score reweighting with relevance user feedback.

The experimental results demonstrate that using GTM semantic similarity improves the clustering results compared to the baseline (Euclidean distance based on TF-IDF). In addition, the Similarity Score Reweighting with Relevance User Feedback strategy achieves the best result and the Internal Centrality-Based Weighting strategy performs better than AVG and MAX, as the first strategy allows users to assign different similarity scores for different elements between two Reddit submissions and the second strategy computes the weight factor of each element of a Reddit submission based on its semantic content.

For future work, we aim to extend this proposed framework to other social network websites, such as Twitter, Facebook, and Google Plus. Another important direction is to extend this framework for studying the spread of rumours in online social networks. For example, visualizing how a rumour-related meme is discussed and spread in Reddit. This will help researchers to understand how a rumour is spread, its pattern and detect emerging rumours. Comparing the spread of rumour-driven memes between Reddit and other OSNs and finding a correlation between them will provide a more holistic view of rumour spread.

Acknowledgment

The research was funded in part by the Natural Sciences and Engineering Research Council of Canada, CNPq, FAPESP, International Development Research Centre, Ottawa, Canada, and Social Sciences and Humanities Research Council of Canada.

References

[1] E. Morozov. (2009) Swine flu: Twitter's power to misinform @ONLINE.
Available: http://neteffect.foreignpolicy.com/posts/2009/04/25/swine_flu_twitters_power_to_misinform [Accessed: April 15, 2015].

[2] FEMA. (2012) Hurricane sandy: Rumor control @ONLINE.
Available: http://neteffect.foreignpolicy.com/posts/2009/04/25/swine_flu_twitters_power_to_misinform [Accessed: April 15, 2015].

[3] M. JafariAsbagh, E. Ferrara, O. Varol, F. Menczer, and A. Flammini, "Clustering memes in social media streams," *Social Network Analysis and Mining*, vol. 4, no. 1, 2014. [Online]. Available: http://dx.doi.org/10.1007/s13278-014-0237-x

[4] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity: Measuring the relatedness of concepts," in *Demonstration Papers at HLT-NAACL 2004*, ser. HLT-NAACL–Demonstrations '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 38–41. [Online]. Available: http://dl.acm.org/citation.cfm?id=1614025.1614037

[5] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou, "Exploiting wikipedia as external knowledge for document clustering," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 389–396. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557066

[6] A. Islam, E. Milios, and V. Kešelj, "Text similarity using google tri-grams," in *Proceedings of the 25th Canadian Conference on Advances in Artificial Intelligence*, ser. Canadian AI'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 312–317. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30353-1_29

[7] A. Strehl, E. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," in *Workshop on Artificial Intelligence for Web Search (AAAI 2000*. AAAI, 2000, pp. 58–64.

[8] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines." *WWW*, vol. 7, pp. 757–766, 2007.

[9] T. Brants and A. Franz, "The google web 1t 5-gram corpus version 1.1," *Technical Report*, 2006.

[10] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proceedings of the Tenth International Workshop on Multimedia Data Mining*. ACM, 2010, p. 4.

[11] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 497–506.

[12] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park, "ivisclustering: An interactive visual document clustering via topic modeling," in *Computer Graphics Forum*, vol. 31, no. 3pt3. Wiley Online Library, 2012, pp. 1155–1164.

[13] J. Choo, H. Lee, Z. Liu, J. Stasko, and H. Park, "An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 865 402–865 402.

[14] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the Twenty-first International Conference on Machine Learning*. ACM, 2004, p. 11.

[15] M. M. Deza and E. Deza, *Encyclopedia of Distances*. Springer, 2009.