Fast Extraction of Statistically Relevant Descriptor Words for Social Media Communities

Arnulfo Azcarraga
[1] Software Technology Dept
College of Computer Studies
De La Salle University
Manila, Philippines
arnie.azcarraga@delasalle.ph

Arces Talavera[1, 2]
[2] Dept of Computer Science
and Information Engineering
National Taiwan University of
Science and Technology
Taipei, Taiwan
arces_talavera@dlsu.edu.ph

***Abstract*—Social media communities can be characterized by descriptor words that are frequently used by its community members but are less often used in other communities. These can be extracted by computing a descriptor index and choosing those words with the highest index. The novel descriptor index proposed here is based on the z-score that measures the frequency of a word in a given community relative to the frequency of the word in all the communities combined, using a statistical standard error. The measure based on z-scores is validated by comparing the words extracted when using z-scores with the words extracted using the fairly popular Term Frequency-Inverse Document Frequency (TF-IDF) and the Lagus method. Once it is established that z-scores can be used to extract descriptor words, the next hurdle is to reduce the dimensionality of the vector space model, where each word that appears in any of the social community messages would constitute one dimension in the vector space model. The solution explored here, used in tandem with z-scores as descriptor index measure, is the Random Projection method. In this dimensionality reduction method, more than 40,000 unique words (dimensions) are randomly projected to as few as 400 dimensions (99% reduction) and yet the proposed scheme still extracts essentially the same descriptor words for each community. To evaluate the combined use of z-scores and Random Projection, and to determine some suitable parameter values for the proper execution of the Random Projection method, 10 communities on Facebook were selected. Despite using only 1% of the original number of dimensions, there is a match of 85% of the top 10 descriptor words between those extracted with all 40,000 dimensions compared to those extracted with only 400.**

Keywords*—Random Projection; Dimensionality Reduction; Social Media; Text Analytics.

I. INTRODUCTION

The emergence of social media allowed the users of the Internet to share their own content and information with others, and to have the opportunity to form their own virtual social network [1]. In fact, social media users are drawn together and are more likely to connect and participate in a social network having people who are popular or people who are similar to them, such as other users who prefer to use the same language as them [2][3].

Characterizing communities in social media networks using descriptor words that are frequently used by the community members is useful when one needs to quickly distinguish one community from another. This can be done by computing a descriptor index measure for each word, and choosing the top *k* words with the best descriptor index measures.

The descriptor index proposed here is based on the z-score that measures the relative frequency of a word in a given community, with a confidence interval, from the frequency of the word in all the communities combined, using a standardized computation of the statistical standard error. The measure based on z-scores is validated by comparing the words extracted when using z-scores with the words extracted using Term Frequency-Inverse Document Frequency (TF-IDF) [4] and the Lagus method based on a goodness measure [5]. Such descriptor words are very useful in Text Mining [6], as well as in other more focused application areas, such as human and social analytics. One simply considers the descriptor words to have a fairly good idea of what the given social media community is concerned with.

Text mining for all its vast potential, however, faces the challenge of having to deal with very large volumes of documents. This in turn translates into a very high dimensional vector space model – where each word that appears in any of the social community documents/messages would constitute one dimension in the vector space. The high dimensionality elevates both the computational and space complexities [7] of any task involving the unique words and phrases that appear in the documents. Indeed, even if we can establish that z-scores can be used to extract descriptor words, the next challenge is to drastically reduce the dimensionality of the vector space model.

Dimensionality reduction is understandably a widely researched area in text mining [7] and other areas, such as image processing, bioinformatics, and so on [8][9]. One method stands out. The Random Projection method [10] randomly projects the high-dimensional features of a dataset into a far smaller low-dimensional space, to as little as just 1% of the original dimensionality of the vector space. And yet, when properly used in combination with other algorithms that are compatible with the projection method, the performance of the low dimensional space is comparable to that in the original, high-dimensional space.

The rest of the paper is organized as follows. Section II describes the characterization of social media communities with a novel bag-of-words representation using z-scores, and validates the proposed approach by comparing the extracted descriptor words to those extracted by TF-IDF and the Lagus method. Section III illustrates how the Random Projection method works, and how it can be used in combination with the z-scores that are computed as descriptor index. Section IV contains the discussion of the experimental results that are designed to fine-tune the parameter values of the Random Projection method. The paper ends with Section V, which contains the conclusions.

II. Characterization of Social Media Communities Using Z-Score

Online social networking applications nowadays implement a “verified user system” to emphasize the importance of opinion leaders, such as celebrities having numerous followers [11]. These leaders are able to make use of the “public reach” provided to them by the social networking site in order to gain lots of attentions and to promote their products and services to their followers.

Every social network community formed by a celebrity and his/her followers often gravitate towards a small subset of words, sometimes transforming to a veritable “jargon”, that over time would be good descriptor words that would characterize the interest, motivation, and even the socio-cultural and linguistic characteristics of the community as a whole.

For the rest of the discussions and experiments in this paper, we use a huge collection of individual “posts” published as public comments on the Facebook pages of 10 well-known Filipino celebrities, having hundreds of thousands or even millions of followers. These posts were collected using Facebook's Graph Application Programming Interface (API) [12].

The steps of preprocessing the dataset include the removal of non-alphanumeric characters, digits, and unnecessary white spaces. In addition, both Filipino [13] and English [14] standard “stop words” in the posts are removed. The remaining tokens/words are then stored using a bag-of-words representation in a vector space. This resulted in a large celebrity dataset containing 40,126 words, where each unique word is considered as an individual “feature”.

A. *Z-score*

The z-scores of each of the words of each of the communities are then computed as their “descriptor index”, and the top *k* words with the highest z-scores in each community are considered to be the descriptor words.

The z-score of a word can be calculated using the following formula:

$$z = \frac{p-P}{SE_P} \tag{1}$$

where *p* is the proportion of the usage of the word relative to all the words in the same community, while *P* is the proportion of the usage of the word relative to all the words in the corpus (in this case, all 10 communities). *SE_P* represents the standard error of the word in the population. This is computed using the following formula:

$$SE_P = \sqrt{\frac{P(1-P)}{n}} \tag{2}$$

where *n* is the number of words in the given community.

Note that the z-score is very commonly used in statistics to test whether the proportion in a given sample is no different from the “ideal” proportion, in this case the population proportion. Candidate descriptor words of a community have proportion *p* that is significantly larger compared to *P* (proportion in all the communities combined).

To test the plausibility of using z-score in extracting the descriptor words of a community, we compute these scores for each word in each of the 10 communities and we extract the top 10 descriptor words. These are shown in Figure 1 as word clouds. Note that the communities have different descriptor words, and when taken together, they give a good idea of the type of discussions that happen in each community. For example, the celebrity “Mocha Uson” has “Duterte (current president of the Philippines)”, “pangulo (president)” and “bayan” (country) as descriptor words. Indeed, she was an active endorser of President Duterte on social media since the campaign period in late 2015. The socio-linguistic characteristics of the celebrities are also apparent. Lea Salonga, Erwan Heussaff, and Anne Curtis have posts that are mostly in English, while the others are mostly in Tagalog. One of the known local celebrities who is known to often be talking of her son has the nickname of her son as one of the extracted descriptor words. Another female celebrity has the name of her husband, also a local celebrity, among the descriptor words.

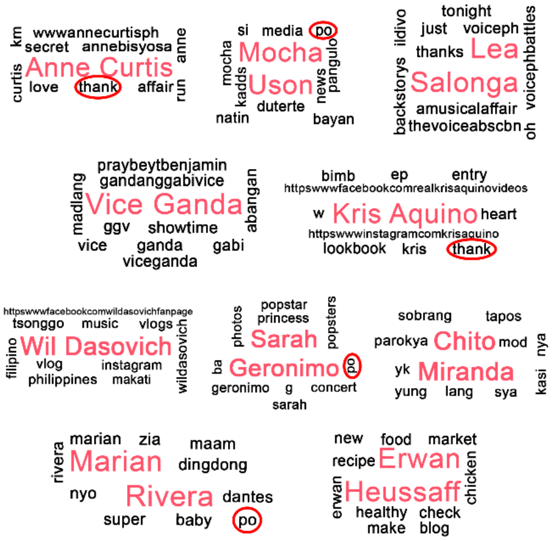

Fig. 1. Top 10 descriptor words for every celebrity, whose names are written in red text, if z-score is used as the measure to characterize each community.

Note that descriptor words need not be unique to a given community. However, when descriptor words are far too common, and appear as descriptor in almost all communities, then they are no longer suitable descriptor words. From Figure 1, we note that using the z-scores kept two terms *po* and *thank*, which are tagged as descriptor words for two celebrity communities. This can be caused by the 2 communities having an extraordinarily high proportion of usage of the terms – such as the Tagalog speakers who use the “po” (as a sign of politeness) in almost every sentence.

B. *TF-IDF*

TF-IDF is the function commonly used to measure the importance of the terms in a given document [15]. This is a standard metric in the vector space model for text mining where the best terms to represent a document are those with high Term Frequency (TF) but at the same time should also be rarely used in other documents - high Inverse Document Frequency (IDF).

In a way, TF alone can be used to extract the descriptor words of a community. It is the proportion of the occurrences of a given term to the entirety of the terms in a given community. The formula for computing TF is as follows:

$$TF_{i,j} = \frac{n_{i,j}}{N_j} \tag{3}$$

where $n_{i,j}$ would be the number of occurrences of term i in document j and N_j represents the totality of the terms in document j.

The suitability of TF as index to extract the descriptor words of the communities is also evaluated. Using TF as the measure in finding the descriptor words of a community is only partially able to achieve its goal. More words, such as *thank, good, si, po, day,* and *watch,* are extracted for a multiple number of communities. This means that TF alone is not able to truly extract descriptor words when we also require that the ideal descriptor words are common in a given community, but relatively less common in the other communities.

IDF [16] is thus introduced to favor terms that are concentrated only in a few documents/communities. This measure ranges from 0, which represents the words that occurred in the entire corpus, to 1, which represents the words that occurred only in one document/community of the entire corpus. The formula for computing IDF is as follows:

$$IDF_i = \log(\frac{D}{d_i}) \tag{4}$$

where D is the total number of documents in the corpus and d_i is the number of documents using the term i.

Given the nature of IDF, it cannot be used as a lone measure to obtain the descriptor words for a community, but it can be applied to retrieve the words that are common in the corpus, or the words that only occur in a handful of communities.

Combining the weights of both the TF and IDF into the TF-IDF measure would indeed improve the extraction of the descriptor words of a document as it combines the importance of the words inside a specific community, and the exceptional words in the 10 communities (entire corpus). Using the TF-IDF removes the terms that are too commonly used, such as *day, thank, love, today, tonight,* and the like, and some of the very common Filipino stop words that are not covered in the stop word list, such as *po, natin, yung, lang,* and such.

C. Goodness Measure

The Lagus method, based on a measure of goodness, describes yet another descriptor word index [5]. It can be observed by ranking the words used in the communities based on the following criteria that: (1) the given word should be more prominent in a community compared to the other words in the same community, and (2) the given word is relatively more prominent in the specified community in the rest of the corpus. The formula for computing the goodness of a characteristic word w given document/community j is as follows [17]:

$$G(w,j) = \mathrm{F_j}(w)\frac{F_j(w)}{\sum_i F_i(w)} \tag{5}$$

where $F_j(w)$ is the proportion of the term w and all the terms in community j. $F_j(w)/\sum_i F_i(w)$ compares the importance of the word w in community j with its importance in the corpus.

D. Comparison of Results

Using the z-score as descriptor index measure can be validated by comparing it to other probability-based measures such as TF, TF-IDF or the Lagus method. A large intersection among the top 10 words when z-score is used as the measure and the top 10 words when the other measures are used would indicate that z-scores are also effective for determining the descriptor index measure of words. The intersections for the communities of each of the chosen celebrities are shown in Table I.

TABLE I. INTERSECTION OF THE TOP 10 WORDS WHEN Z-SCORE IS USED AND THE TOP 10 WORDS WHEN OTHER MEASURES ARE USED

Celebrity Community	*TF*	*TF-IDF*	*Goodness*
Anne Curtis	5	4	8
Chito Miranda	6	3	9
Erwan Heussaff	8	2	8
Kris Aquino	7	6	9
Lea Salonga	3	5	8
Marian Rivera	8	6	9
Mocha Uson	7	4	8
Sarah Geronimo	8	5	7
Vice Ganda	9	5	7
Wil Dasovich	8	5	6
AVERAGE	6.9	4.5	7.9

Using the z-score as the measure to find the descriptor words of the communities yields an average of 7.9 out of the top 10 words when compared to using the Lagus method, and averages of 6.9 and 4.5 when compared to TF and IDF, respectively. Much like when evaluating alternative methods that recognize specific objects from images [18][19] (which just need to list the correct object somewhere within the top 5 predicted objects in an image), we do not require that the top descriptor word in each community, that are extracted using z-scores, would be the same descriptor words extracted when using TF, TF-IDF and the Lagus method. Since there is no real “ground truth” as to which are the real top 10 descriptor words, we would simply need to see that there is some fair amount of intersection among the extracted descriptor words when compared to the other methods.

We also compare the results yielded when using TF to TF-IDF and the goodness measure (Table II). Compared to using z-scores, TF is only able to get an average of 2.9 out of the top 10 words when compared to the TF-IDF measure, and an average of 5.6 out of the top 10 words for the goodness measure. Lastly, we evaluate the Lagus method by comparing it to the TF-IDF measure in extracting the top descriptive words of the celebrities. The results are also presented in Table II.

TABLE II. INTERSECTION OF THE TOP 10 WORDS BETWEEN THE VARIOUS PROBABILITY-BASED MEASURES

Celebrity Community	*TF vs. TF-IDF*	*TF vs. Goodness*	*Goodness vs. TF-IDF*
Anne Curtis	2	3	6
Chito Miranda	1	6	4
Erwan Heussaff	1	7	3
Kris Aquino	4	6	7
Lea Salonga	0	2	7
Marian Rivera	5	8	6
Mocha Uson	2	5	6
Sarah Geronimo	5	7	6
Vice Ganda	5	7	7
Wil Dasovich	4	5	7
AVERAGE	2.9	5.6	5.9

The goodness measure of the Lagus method can extract an average of 5.9 out of the top 10 words when compared to the TF-IDF, which has a similar performance as the z-score. Overall, with all the comparisons made, the statistical capabilities of the z-score measure in extracting the descriptive words of the celebrity communities make it a suitable descriptor index measure.

III. APPLICATION OF RANDOM PROJECTION

The general idea of Random Projection [10] is to project the large number of dimensions of a text corpus (in this case, the single vector space for the 10 social communities) into a much smaller vector space with the number of features being a very small fraction of the original number. Random Projection, as depicted in (6),

$$D(i, j) \rightarrow D'(i, m) \qquad (6)$$

transforms vector space D having i documents (messages/communities), and j unique words each of which is a dimension in D, into vector space D' having the same i documents (communities), and m features, where $m << j$.

The projection takes two parameters, m and r. The first parameter m is the (much smaller) number of features of D', and r represents the number of times a given word (feature) is mapped into any of the m dimensions in the projected space D'.

Random projection uses a j x m projection matrix, where every feature in D corresponds to a row in R. The m columns are the components that act as the new features found in D', such that each column in $D'$is in fact the sum of frequencies of a very large number of randomly selected words. Each word in turn is randomly mapped by R to r different columns in D'. At r = 5 or more, there is a small chance that two unique words would have been mapped by R to exactly the same 5 columns in D'.

Constructing the projection matrix R consists of randomly selecting r unique dimensions in m for each of the j words. A value of 1 is given to each of the r selected components, while all remaining $m - r$ dimensions of row j are set to 0. The values for every document/community in D are computed as follows:

$$d'[f_1 \dots f_m] = d[f_1 \dots f_j] * \begin{bmatrix} R_{11} & \cdots & R_{1m} \\ \vdots & \ddots & \\ R_{j1} & & R_{jm} \end{bmatrix}^T \qquad (7)$$

where d' is the resultant, compressed vector in D' for each document d in D.

This Random Projection method thus yields a highly compressed and compact dataset, where each dimension encodes the frequencies of a very large number of words. Note that by the nature of Random Projection, it is not straightforward to choose from m dimensions those that might point us to the suitable set of descriptor words. And, since each of the m dimensions would have mixed up the term frequencies of thousands of words, Random Projection would not be compatible with approaches such as the TF-IDF, which relies heavily on the "rarity" of occurrence of candidate words in documents/communities other than the one for which a descriptor word is being searched.

In the rest of this paper, we go back to z-scores and demonstrate that it is feasible to use it in combination with the Random Projection method – and thus benefit from the ability of the latter to significantly lower the number of dimensions and improve on the time and space complexity of the over-all approach.

Since a z-score relies heavily on the TF except that it also incorporates a test for statistically significant differences in proportions between candidate words, we now show how z-scores can still be used to select descriptor words, even if we drastically reduce the dimensions of our vector space.

All the words in the original dataset are first given the projected z-score value based on the summation of the z-scores of the components in D' where they are projected. As an example, given that the word f_1 is projected into the components m_0, m_3, m_5, and m_{10}, then w is given the value of the summation of the z-scores of the stated components. The process, as shown in Figure 2, is done for all words in the original dataset.

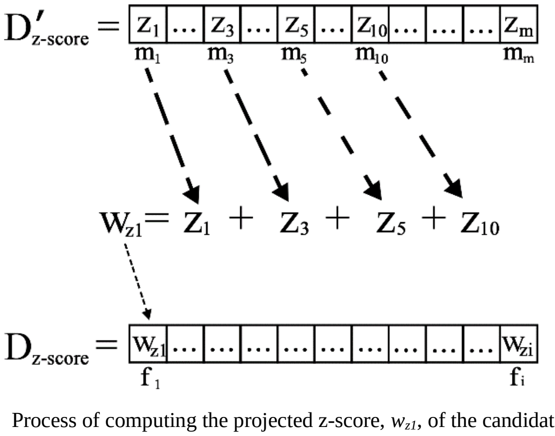

Fig. 2. Process of computing the projected z-score, w_{z1}, of the candidate descriptor word f_1. The z-scores z_1, z_3, z_5, and z_{10} of the components where f_1 is projected are added to determine its projected z-score.

IV. Random Projection Based on Z-Scores

A. Filtering of Riders

Because of how Random Projection works, we can expect that certain words can be mistakenly selected as a descriptor word simply because in the projection they happen to be mapped to the same dimensions where the real descriptor words get mapped to as well. These words are what we call *riders* - that piggy-back on the real descriptor words and thus get high scores for themselves.

Removing such *riders* turns out to be straightforward. We simply apply the Random Projection *k* times, and words that do not get consistently high scores are plain *riders*. The final scores of the words are computed by taking the average of their projected scores in all of the *k* trials.

A maximum of 15 Random Projection trials are performed to test if the performance is affected by the number of trials done. Parameter values for the projection matrix used are m = 400 and r = 10. The performance is measured by the number of words that are extracted using only m dimensions that match those extracted words when using all 40,000 dimensions and the comparison is presented in Table III.

TABLE III. Comparison of the performances of the various number of times Random Projection is applied in getting the top words of the non-projected dataset

Number of Trials	*Top 5 Words*	*Top 10 Words*
1	66%	43%
2	76%	68%
3	80%	76%
4	76%	75%
5	80%	84%
6	82%	85%
7	80%	85%
8	82%	86%
9	82%	87%
10	82%	88%
11	84%	88%
12	84%	87%
13	82%	88%
14	84%	89%
15	82%	88%

Performing Random Projection twice already significantly improves its performance from 66% to 76% among top 5 words, and 43% to 68% among top 10 words, when compared to only having done Random Projection once. This shows that the *riders* are easily removed. And, as the number of trials increases, the performance rate of Random Projection improves further. From 9 trials onwards, the performance rate stabilizes between 82% to 84% when taking the top 5 words, while for taking the top 10 words, performance is between 87% to 89%. Since we would not want to make more trials (runs) than what is necessary, we just use 10 trials for the remaining experiments that needed to be conducted to further fine-tune the method.

B. Improving Random Projection Results

We proceed to further improve the performance of the approach by encoding the z-scores before they are accumulated during Random Projection, such as taking the $\log_{10}$ value, its square root, or by squaring it. Taking the $\log_{10}$ or the square root of the z-scores tends to diminish the differences among z-scores, while taking the squared-value would obviously increase the effect of the differences among z-scores (which are mostly numbers greater than 1.0).

In the following experiments, both m and r parameters of Random Projection are given the same old values of 400 and 10, respectively. To evaluate the performance of the techniques applied, we again use the intersection of the top 5 and top 10 descriptor words using the raw non-projected z-scores (all 40,000 dimensions are used) and the projected raw or encoded z-scores. The comparison of performances for the different measures applied is shown in Table IV.

TABLE IV. Comparison of the performances of the various encodings of the z-scores used as the basis

Z-Score Representation	*Top 5 Words*	*Top 10 Words*
raw z	56%	33%
$\log_{10}$ z	62%	38%
$\sqrt{z}$	66%	43%
z^2	63%	19%

Indeed, adding encoding to the z-scores has a noticeable effect on the performance of Random Projection. The baseline performance rate of using the raw z-score is 56% when getting the top 5 words, and 33% when getting the top 10 words. Table IV shows that the performance worsens when we use the squared values of the z-scores. Upon inspection, we noticed that, what happened in face was that, when we square the z-scores, the *riders* increase in rank and get selected. Indeed, squaring the z-scores yielded a performance rate of only 36% when taking the top 5 words, and 19% for the top 10 words.

Table IV also shows that, when we diminish the importance of the differences among z-scores by getting the $\log_{10}$ or their square root values, we see an improvement in the performance rate of Random Projection. Using $\log_{10}$ yields performance rates of 62% and 38% when getting the top 5, and 10 words, respectively. Taking the square root of the z-scores has the best performance rates of 66% for the top 5 words, and 43% for the top 10 words.

The performance rates can be even further improved. To fully utilize the capabilities of Random Projection, the next step is to try to find good values for the two parameters: r, the number of times the features will be projected, and m, the number of features of the projected, reduced dataset.

Various combinations of the two parameters, m and r are evaluated. The values of m start with 100, which is then doubled until the value reaches 3,200 (roughly 8% of the original number of dimensions). For r, the experiments start with a value of 5. In this paper, we only show the results for r having values 5, 10, 20, 40, and 80.

This time, 10 trials of Random Projection are done (to remove the riders) and the square root representation of the z-scores are used, at a fixed value of $m = 400$. The comparison of the performances of the different values of r is shown in Table V.

TABLE V. COMPARISON OF THE PERFORMANCES OF THE VARIOUS VALUES OF THE *R* PARAMETER OF RANDOM PROJECTION

Value of r	*Top 5 Words*	*Top 10 Words*
5	84%	85%
10	82%	88%
20	80%	86%
40	74%	85%
80	84%	85%

From Table V, it can be observed that parameter r can be set to low values of 10, or even 5, and yet, good results of over 80% can be achieved. We are left with now finding the value of *m*, which was set to 400 in the experiments earlier.

The final experiment evaluates the performance of Random Projection when the *m* parameter is given differing values. For the remaining experiments, 10 trials and square root encoding are used, with the *r* parameter set to 5. The comparison of the performance rates is shown in Table VI.

TABLE VI. COMPARISON OF THE PERFORMANCES OF THE VARIOUS VALUES OF THE *M* PARAMETER OF RANDOM PROJECTION

Value of m	*Top 5 Words*	*Top 10 Words*
100	80%	71%
200	86%	79%
400	84%	85%
800	76%	88%
1600	88%	91%
3200	90%	94%

As the value of *m* increases, the performance rate of Random Projection also understandably improves. But since the aim of dimensionality reduction is to get good results in significantly less amount of time, we would prefer lower values for *m*. At $m = 3{,}200$, the performance rate is 90% when taking the top 5 descriptor words, and 94% when taking the top 10 words. Dramatically reducing the number of dimensions to even just 10% of its original size, at $m = 400$, Random Projection still yields a decent performance of 84% and 85% when taking the top 5 and top 10 descriptor words, respectively.

V. CONCLUSION AND FUTURE WORK

The computation of a descriptor index to extract the descriptor words that characterize social media communities can indeed be based on the z-score that measures the relative frequency of a word in a given community compared to its frequency in all the communities combined.

This novel measure is validated using a collection of "posts" published by 10 well-known Filipino celebrities on their Facebook pages. The words extracted using the z-scores are compared with the words extracted using the TF, TF-IDF, and the Lagus method based on the goodness measure to establish that z-scores can be effective as basis for extracting descriptor words.

The other challenge addressed in this paper is the reduction of the dimensionality of the vector space using the Random Projection method. This dimensionality reduction method randomly projects more than 40,000 unique words to as few as 400 dimensions. This method is used alongside the z-scores as descriptor index measure to accurately and efficiently extract descriptor words for each community.

The combination of z-scores and Random Projection is evaluated using the celebrity dataset. Reducing the dimensionality of the original 40,000 dimensions to only 8% of it (3,200 dimensions) yields a 94% match among top 10 descriptor words compared to using all 40,000 dimensions. Pushing the dimensionality reduction further, using only 1% of the original number of dimensions produces a match of 85% of the top 10 descriptor words compared to those extracted using all 40,000 dimensions. For further benchmarking and validation of the approach, future experiments may concentrate on even larger social media communities, preferably those using English, Spanish, or French, so that a larger comparison and evaluation among alternative techniques may be conducted.

REFERENCES

[1] P. M. Napoli, “Social media and the public interest: Governance of news platforms in the realm of individual and algorithmic gatekeepers,” *Telecommunications Policy*, vol. 39, no. 9, 2015, pp. 751 – 760.

[2] P. Saha and R. Menezes, “A language-centric study of twitter connectivity,” in *Social Informatics*, E. Spiro and Y.-Y. Ahn, Eds. Cham: Springer International Publishing, 2016, pp. 485–499.

[3] P. Saha, *Language relations on twitter: A network science approach*. PhD [Dissertation]. Melbourne, FL: Florida Tech., April 2017. [Online]. Available: http://hdl.handle.net/11141/1417 [Retrieved April 2018].

[4] G. Salton and C. T. Yu, “On the construction of effective vocabularies for information retrieval,” in *Proceedings of the 1973 Meeting on Programming Languages and Information Retrieval*, ser. SIGPLAN ’73. New York, NY, USA: ACM, 1973, pp. 48–60.

[5] K. Lagus and S. Kaski, “Keyword selection method for characterizing text document maps,” in *Proceedings of the 1999 International Conference on Artificial Neural Networks*. IET, 1999, pp. 371–376.

[6] Y. Zhang, M. Chen and L. Liu, "A review on text mining," in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Sept 2015, pp. 681-685.

[7] G. S. Reddy, “Dimensionality reduction approach for high dimensional text documents,” in *2016 International Conference on Engineer & MIS (ICEMIS).* Sept 2016, pp. 1-6.

[8] W. Awada, T. M. Khoshgoftaar, D. Dittman, R. Wald and A. Napolitano, "A review of the stability of feature selection techniques for bioinformatics data," in *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, Las Vegas, NV, 2012, pp. 356-363. doi: 10.1109/IRI.2012.6303031

[9] D. L. Padmaja and B. Vishnuvardhan, “Comparative study of feature subset selection methods for dimensionality reduction on scientific data,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Feb 2016, pp. 31–34.

[10] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, “Latent semantic indexing: A probabilistic analysis,” in *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ser. PODS ’98. New York, NY, USA: ACM, 1998, pp. 159–168.

[11] W. X. Zhao, J. Liu, Y. He, C. Y. Lin, and J. R. Wen, “A computational approach to measuring the correlation between expertise and social

media influence for celebrities on microblogs," in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, Aug 2014, pp. 460–463.

[12] Facebook. The graph api. [Online]. Available: https://developers. facebook.com/docs/graph-api/ [Retrieved Nov. 2017].

[13] G. Diaz. Stopwords iso. [Online]. Available: https://github.com/stopwords-iso/stopwords-tl/blob/master/stopwords-tl.txt [Retrieved Nov. 2017].

[14] University of Glasgow. Stop word list. [Online]. Available: http: //ir.dcs.gla.ac.uk/resources/linguistic utils/stop words [Retrieved Nov. 2017].

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage*., vol. 24, no. 5, pp. 513–523, Aug. 1988.

[16] K. Sparck Jones, "Document retrieval systems," P. Willett, Ed. London, UK, UK: Taylor Graham Publishing, 1988, ch. A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pp. 132–142.

[17] K. Lagus, S. Kaski, and T. Kohonen, "Mining massive document collections by the websom method," *Inf. Sci.*, vol. 163, no. 1-3, pp. 135–156, Jun. 2004.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.