A Novel Word Embedding Based Stemming Approach for Microblog Retrieval during Disasters

Moumita Basu[1,2], Anurag Roy[1], Kripabandhu Ghosh[3], Somprakash Bandyopadhyay[2], and Saptarshi Ghosh[1,4]

[1] Indian Institute of Engineering Science and Technology, Shibpur, India
[2] Indian Institute of Management, Calcutta, India
[3] Indian Institute of Technology, Kanpur, India
[4] Indian Institute of Technology, Kharagpur, India
saptarshi@cse.iitkgp.ernet.in

Abstract. IR methods are increasingly being applied over microblogs to extract real-time information, such as during disaster events. In such sites, most of the user-generated content is written informally – the same word is often spelled differently by different users, and words are shortened arbitrarily due to the length limitations on microblogs. Stemming is a common step for improving retrieval performance by unifying different morphological variants of a word. In this study, we show that rule-based stemming meant for formal text often cannot capture the arbitrary variations of words in microblogs. We propose a context-specific stemming algorithm, based on word embeddings, which can capture many more variations of words than what can be detected by conventional stemmers. Experiments on a large set of English microblogs posted during a recent disaster event shows that, the proposed stemming gives considerably better retrieval performance compared to Porter stemming.

Keywords: Microblog Retrieval, Stemming, Disasters, Word Embedding, word2vec

1 Introduction

In recent years, microblogging sites (e.g., Twitter, Weibo), have become important sources of information on various topics and events, and Information Retrieval (IR) on microblogs (tweets) is now an important area of research. In such forums, the user-generated content is often written in informal, casual ways. Additionally, due to the strict limitation on the length of microblogs (140 characters at most), words are often abbreviated arbitrarily, i.e., without obeying any linguistic rules. Such arbitrary variations of words negatively affect the performance of IR methods. This factor is especially crucial in situations such as a disaster event (earthquake, flood, etc.), when it is important to retrieve all relevant information irrespective of the word variations.

In such scenarios, IR methods usually rely on stemming algorithms (stemmers), whose purpose is to improve retrieval performance by mapping the morphological variants (usually inflectional) of a word to a common *stem*. Some stemmers are language-specific (e.g., the popular Porter stemmer [7] for English), utilizing the rules of a natural language to identify variants of a word; while there has been language-independent algorithms also [4,5]. However, all the prior contributions in this area have been on text

Variations	Excerpts from tweets
Variations in spellings of a word	
epicentre	6.7 magnitude #earthquake **epicentre** 49 km from Banepa in #Nepal says USGS
epicenter	5.0 earthquake, 29km SSW of Kodari, Nepal. Apr 26 13:11 at **epicenter**
gurudwara	Delhi Sikh **Gurudwara** committee will send 25k food packet everyday to nepal
gurdwara	Delhi Sikh **Gurdwara** body to send Langar (food) for Nepal earthquake hit people
sindhupalchowk	**#Sindhupalchowk** 1100+deaths and 99% Houses are Down
sindhupalchok	Indian national Azhar 23, missing. Last location **Sindhupalchok**. Plz help.
sindupalchowk	Food Distribution in **sindupalchowk**, sufficient for 7 days for 500 victims
dharhara	Earthquake in Nepal: 180 bodies retrieved from **Dharhara** tower debris
dharahara	Historic **Dharahara** Tower in #Kathmandu, has collapsed #earthquake
dharara	**Dharara** Tower, built in 1832, collapses in #Kathmandu during earthquake
Arbitrarily shortened forms of a word	
building	Earthquake destroyed hospital, road, **building** in Kavre district of Nepal.
bldg	Nepal quake stresses importance of earthquake resistant **bldg** designs in entire NCR.
secretary	Foreign **Secretary** statement on #Nepal earthquake available here [url]
secy	Foreign **Secy** & Defence **Secy** giving latest updates on earthquake relief [url]
medical	India: NDRF personnel, crack **medical** team with relief rushed to #Nepal
med	4 planes to leave for #Nepal tmrw carry **meds**, **med** team, 30-Bed Hospital
IndianAirForce	**#IndianAirForce** / Army already helping with relief, food, medicines, & all calls to Nepal subsided
IAF	Drinking water plus emergency relief supplies headed to #Nepal by **#IAF** aircraft AP Photo [url]

Table 1. Examples of morphological variations of words, and tweets containing the variations (from a collection of tweets related to the Nepal-India earthquake in April 2015).

written in a formal way, and stemming on such informal, noisy text (like microblogs) has not been studied well in literature.

Motivation: We motivate the need for new stemming algorithms for microblogs through a case study. We collected a large set of microblogs posted during a particular disaster event – the earthquake in Nepal and India in April 2015 (see Section 3 for details of the dataset). We observed different variations of many words in this collection of microblogs, some of which are shown in Table 1. We found two broad types of variations: (1) *Different spellings of a word:* Several words are spelled differently by different users; such words include both English words (like 'epicentre' and 'epicenter') as well as non-English words (like 'gurudwara' and 'gurdwara'). Also, proper nouns like names of places are often spelled differently (e.g., 'Sindhupalchowk', 'Sindhupalchok', and 'Sindupalchowk'), as shown in Table 1. (2) *Arbitrarily shortened forms of a word:* Due to the strict limitation on the length of microblogs, words are often shortened arbitrarily, e.g., 'building' shortened to 'bldg', 'medical' shortened to 'med', and so on (see Table 1). Such variations do not conform to rules of the English language, and hence cannot be identified by standard stemmers like the Porter stemmer.

In this work, we propose a context-specific stemming algorithm that can identify arbitrary morphological variations of words in a given collection of microblogs. We view this as a stemming problem because we assume that the variations of a word will share some common initial characters (a *common prefix*). However, we consider this common prefix to be very short, preferably shorter than 3 characters (which was advocated by Paik *et al.* [5] for formally written text).

Note that stemming algorithms that use common prefix length with word association [5] has been found to out-perform the ones using only common prefix length [2]. The role of *context* becomes particularly important in the case of microblogs where non-standard word representations are ubiquitous (as evident from Table 1). Here, only a combination of common prefix and context can possibly group semantically related variants. In this work, we use the word-embedding tool word2vec [3] to harness the context of word variants, in conjunction with common prefix length and other string simi-

larity measures, to identify inflectional variants of words. We compare retrieval performance over differently stemmed versions of a collection of English microblogs posted during the Nepal earthquake – using Porter stemming, and our proposed stemming – and demonstrate that our proposed stemming algorithm yields statistically significantly better retrieval performance for the same queries over microblogs.

2 Proposed stemming algorithm for microblogs

This section describes our proposed stemming algorithm for microblogs. For a given collection of microblogs, let $\mathbb{L}$ be the lexicon (i.e., the set of all words case-folded to lower case) of the collection, excluding the stopwords, urls, user_mentions, email-ids, and other non alpha-numeric words. We first describe how we identify 'similar' words, and then describe the stemming algorithm.

2.1 Measuring word similarity

To judge if two words w, $w^* \in \mathbb{L}$ are similar (i.e., likely variants of one another), we consider two basic types of similarity between the words, as follows.

(1) String similarity: This is checked in two steps. First, we check if the two words have a common prefix of length p (p is a positive integer). We consider p to be very short for informal, noisy microblogs, specifically $p \leq 3$, since a common prefix of length 3 was advocated in [5] for *formally written* text. Second, we calculate the length of the Longest Common Subsequence [1] of the two words, denoted by $LCS_{length}(w, w^*)$.

(2) Contextual similarity: We trained word2vec [3] over the set of tweets.[5] The word2vec model gives a vector for each term in the corpus, which we refer to as the *term-vector*. Let $\overrightarrow{w}$ and $\overrightarrow{w^*}$ be the word2vec *term-vectors* of w and w^* respectively. The term-vector is expected to capture the context in which a word is used in the corpus [3]. Hence, contextual similarity of the two words is quantified as the cosine similarity of the corresponding word2vec term-vectors, denoted as $cos\_sim(\overrightarrow{w}, \overrightarrow{w^*})$.

Thus, we consider w^* to be a likely variant of w only if they have sufficient string similarity, as well as they have been used in a similar context in the corpus.

2.2 Proposed stemming algorithm

Our proposed stemming algorithm has the following two phases.

Phase 1: Identifying possible variants of words: This phase is aimed at identifying the possible variants of words on the basis of *string similarity*, as described above. For each word $w \in \mathbb{L}$, we construct a set L_w that contains all the words $w^* \in \mathbb{L}$ satisfying the following three conditions: (1) w^* has the same common prefix of length p as w, where $p \leq 3$. (2) $|w^*| \leq |w|$, i.e., w^* is of length less than or equal to length of w. We consider this condition because w^* is supposed to be a stem of w and conventionally a stem is smaller or equal in length as the original word. (3) The length of the Longest Common Subsequence of characters between w and w^*, $LCS_{length}(w, w^*) \geq \alpha|w^*|$, where $\alpha \in [0, 1]$ is a parameter of the algorithm. This condition ensures that the variants

[5] The Gensim implementation for word2vec was used – `https://radimrehurek.com/gensim/models/word2vec.html`. The continuous bag of words model is used for the training, along with Hierarchical softmax, with the following parameter values – Vector size: 2000, Context size: 5, Learning rate: 0.05.

Group of words stemmed to a common stem	Stem
contribute, contributed, contribution, contributions	contribute
donating, donate, donated, donates, donation, donations	donate
collapse, collapsing, collapses, collapsed	collapse
gurudwaras, gurudwara, gurdwaras, gurdwara	gurdwara
organisations, organizations, organisation, organization, orgs, org	org
medical, medicine, medicines, medics, meds, med	med

Table 2. Examples of groups of words which were stemmed to a common stem, by the proposed stemming algorithm.

of w have a common subsequence of at least a certain length with w. Thus, L_w contains the possible variants of w.

Phase 2: Identifying the stem: In this phase, we look to filter out those variants of w from the set L_w which have high *contextual similarity* with w. We define the *Stemming Score* ($Stem_{score}$) between w and $w^* \in L_w$ as follows:

$$Stem_{score}(w, w^*) = \beta * cos\_sim(\overrightarrow{w}, \overrightarrow{w^*}) + (1 - \beta) * LCS_{length}(w, w^*) \quad (1)$$

where, $\beta \in [0, 1]$ is another parameter of the algorithm. Note that $Stem_{score}(w, w^*)$ is a measure of both the contextual similarity and string similarity of w and w^*. We choose only those $w^* \in L_w$ as the candidate stems for w for which $Stem_{score}(w, w^*) \geq \gamma$, where $\gamma \in [0, 1]$ is another algorithmic parameter.

We construct a set L_w^s of candidate stems of w, comprising of only those words from L_w which satisfy the above condition ($L_w^s \subseteq L_w$). In case there are multiple words in L_w^s, the word in L_w^s with the *minimum length* is chosen as the stem for the set $\{w\} \cup L_w^s$ (in case of ties, we break ties arbitrarily).

Parameters of the algorithm: The proposed stemming algorithm has four parameters – (i) p, the length of the common prefix ($p \leq 3$), (ii) α, a threshold on the string similarity, (iii) β, which decides the relative importance between string similarity and contextual similarity, and (iv) γ, the final threshold for considering a word as a candidate stem of another. We considered these parameters in order to make the algorithm generalizable for different types of text. The parameters can be decided based on factors such as, how noisy the text is, and how aggressively one wants to identify variants of a word.

Sample output of the algorithm: The algorithm identifies *groups of similar words which are stemmed to a common stem*, some examples of which are shown in Table 2. We see that the algorithm correctly identifies different types of word variations, including variations made following rules of English (e.g., 'donating', 'donated', 'donates' all stemmed to 'donate'), variations in spelling (e.g., 'gurudwara' and 'gurdwara', 'organisations' and 'organizations'), and arbitrarily shortened forms of words (e.g., 'organisations' and 'orgs', 'medicines' and 'meds', etc). Evidently, standard stemmers will not be able to identify many of these variants.

3 Experiments and Results

We now apply the proposed stemming algorithm over a collection of microblogs, and report retrieval performance.

Microblog collection: We consider a large collection of about 100K English tweets posted during a recent disaster event – the Nepal-India earthquake in April 2015.[6] After

[6] https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake

removing duplicate tweets, using a simplified version of the methodologies in [10], we obtained a set of 50,068 tweets. We pre-processed this set of tweets by removing a standard set of English stopwords, URLs, user-mentions, punctuation symbols, and case-folding. All experiments were performed over this pre-processed set of tweets.

Queries and gold standard results: We consulted members of an NGO (Doctors For You) who participated in relief operations during the Nepal earthquake, to know the information requirements during the operations. They suggested some information needs – like resource needs (e.g., medicines, tents, medical teams), damage caused (e.g., damaged houses), the situation at specific locations, etc. – based on which 15 queries were formed as shown in Table 3.

To develop the gold standard, three human annotators were employed to find out the relevant tweets for each query. The annotators were asked to start with the query, observe the matching tweets to identify various variations of the terms in the query, and then retrieve tweets with the variations (if any) as well.

Parameter setting for the present study: As stated earlier, we consider $p \leq 3$ for noisy microblogs. We experimented with $p = 1$ and $p = 2$. Evidently, more variants of words can be identified for $p = 1$; e.g., variants like 'IndianAirForce' and 'IAF', 'building' and 'bldg' can only be identified for $p = 1$. However, such a low value of p also has the risk of identifying false positives. Hence, in this study, we use $p = 2$ for the experiments.

The values of (α, β, γ) were determined using grid search over $\{0.5, 0.6, ..., 1.00\}$ $\times$ $\{0.5, 0.6, ..., 1.00\}$ $\times$ $\{0.5, 0.6, ..., 1.00\}$. The best performance values for our dataset were obtained for $\alpha = 0.6, \beta = 0.7, \gamma = 0.7$, which are being reported.

Baselines: Since we focus on English tweets, we compared our proposed approach with the Porter stemmer [7], perhaps the mostly widely used English language-specific stemming algorithm. We also compare with an unstemmed retrieval performance.

Evaluation measures: We report the retrieval performance in terms of Average Precision (AP) and Recall@1000, for the queries as selected above.

Evaluation results: We used the well-known Indri system [9] which uses a Language Modelling framework [6] for retrieval. Table 3 shows the query-wise retrieval performance over three versions of the microblog collection – unstemmed, Porter stemmed, and stemmed using the proposed approach.

Both Porter stemmer and the proposed stemmer enable better retrieval, than without stemming. Importantly, the proposed stemming approach gives statistically significantly better performance at 95% confidence level (p-value < 0.05) by Wilcoxon signed-rank test [8], than Porter stemming, for both the measures (22% better in terms of MAP, and 18.5% better in terms of Recall@1000, compared to Porter stemmer).

Evidently, the superior performance of the proposed methodology is because the proposed approach is able to identify arbitrary spelling variations which Porter stemmer (based on English language rules) could not identify.

4 Conclusion

We demonstrate that traditional rule-based stemming fails to identify many informal variations of words in microblogs. We also propose a novel context-specific stemming algorithm for microblogs, which takes into account both string similarity and contextual

Query	Average Precision			Recall@1000		
	Unstemmed	Porter	Proposed	Unstemmed	Porter	Proposed
food send	0.1251	0.2356	**0.2542**	0.6214	**0.9660**	0.9563
food packet distributed	0.1930	0.2283	**0.2645**	0.9515	**0.8835**	0.8350
house damage collapse	0.0065	0.0254	**0.0296**	0.2264	0.5283	**0.6226**
medicine need	0.2029	**0.3528**	0.1390	0.4561	0.6140	**0.9298**
tent need	0.1110	**0.5962**	0.5718	0.5195	0.9870	**1.0000**
medicine medical send	0.1806	0.2851	**0.3775**	0.8333	**0.9808**	0.9744
Sindhupalchok	0.4457	0.4457	**0.9493**	0.4457	0.4457	**0.9620**
medical treatment	**0.8003**	0.7998	0.7417	0.8471	0.8471	**1.0000**
medical team send	0.5506	0.7358	**0.7548**	0.9290	0.9484	**0.9935**
NDRF operation	0.7337	0.9006	**0.9065**	0.9653	0.9653	**0.9722**
rescue relief operation	0.5342	0.7205	**0.7440**	0.5846	0.8338	**0.9154**
relief organization	0.2405	0.3015	**0.3293**	0.3448	**0.5460**	0.4598
Dharahara collapse	0.2659	0.6424	**0.9599**	0.7692	0.7692	**0.9780**
epicentre	0.3613	0.3612	**0.9847**	0.3621	0.3642	**0.9853**
gurudwara meal	0.2067	0.6116	**0.8429**	0.2671	0.7671	**0.9795**
All	0.3305	0.4828	**0.5900** (+78.5%, +22.2%)	0.6082	0.7631	**0.9042** (+48.7%, 18.5%)

Table 3. Comparison of the proposed method with Porter stemmer. Retrieval performance reported on three versions of the microblog collection – *unstemmed*, *Porter stemmed*, and stemmed using the proposed approach. The proposed approach significantly outperforms Porter stemmer ($p < 0.05$) in both the measures. Percentage improvements over *unstemmed* and *Porter stemmed* are also shown.

similarity among words. Through experiments on a collection of English microblogs posted during a disaster event, we demonstrate that the proposed stemming algorithm yields much better retrieval performance over the commonly used Porter stemmer.

Acknowledgement: This research was partially supported by a grant from the Information Technology Research Academy (ITRA), DeITY, Government of India (Ref. No.: ITRA/15 (58)/Mobile/DISARM/05).

References

1. Cormen, T., Leiserson, C., Rivest, R., C.Stein: Introduction to Algorithms. The MIT Press, 3rd edn. (2009)
2. Majumder, P., Mitra, M., Parui, S.K., Kole, G., Mitra, P., Datta, K.: Yass: Yet another suffix stripper. ACM Trans. Inf. Syst. 25(4) (Oct 2007)
3. Mikolov, T., Yih, W., Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. In: NAACL HLT 2013 (2013)
4. Paik, J.H., Mitra, M., Parui, S.K., Järvelin, K.: Gras: An effective and efficient stemming algorithm for information retrieval. ACM Trans. Inf. Syst. 29(4), 19:1–19:24 (Dec 2011)
5. Paik, J.H., Pal, D., Parui, S.K.: A novel corpus-based stemming algorithm using co-occurrence statistics. In: Proc. ACM SIGIR. pp. 863–872 (2011)
6. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proc. ACM SIGIR. pp. 275–281 (1998)
7. Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
8. Siegel, S.: Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill series in psychology, McGraw-Hill (1956)
9. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proc. ICIA. Available at: `http://www.lemurproject.org/indri/` (2004)
10. Tao, K., Abel, F., Hauff, C., Houben, G.J., Gadiraju, U.: Groundhog Day: Near-duplicate Detection on Twitter. In: Proc. World Wide Web (WWW) (2013)