Tweet Normalization with Syllables

Ke Xu
School of Software Eng.
Beijing U. of Posts & Telecom.
Beijing 100876, China
xxukez2@gmail.com

Yunqing Xia
STCA
Microsoft
Beijing 100084, China
yxia@microsoft.com

Chin-Hui Lee
School of Electr. & Comp. Eng.
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA
chl@ece.gatech.edu

Abstract

In this paper, we propose a syllable-based method for tweet normalization to study the cognitive process of non-standard word creation in social media. Assuming that syllable plays a fundamental role in forming the non-standard tweet words, we choose syllable as the basic unit and extend the conventional noisy channel model by incorporating the syllables to represent the word-to-word transitions at both word and syllable levels. The syllables are used in our method not only to suggest more candidates, but also to measure similarity between words. Novelty of this work is three-fold: First, to the best of our knowledge, this is an early attempt to explore syllables in tweet normalization. Second, our proposed normalization method relies on unlabeled samples, making it much easier to adapt our method to handle non-standard words in any period of history. And third, we conduct a series of experiments and prove that the proposed method is advantageous over the state-of-art solutions for tweet normalization.

1 Introduction

Due to the casual nature of social media, there exists a large number of non-standard words in text expressions which make it substantially different from formal written text. It is reported in (Liu et al., 2011) that more than 4 million distinct out-of-vocabulary (OOV) tokens are found in the Edinburgh Twitter corpus (Petrovic et al., 2010). This variation poses challenges when performing natural language processing (NLP) tasks (Sproat et al., 2001) based on such texts. Tweet normalization, aiming at converting these OOV non-standard words into their in-vocabulary (IV) formal forms, is therefore viewed as a very important pre-processing task.

Researchers focus their studies in tweet normalization at different levels. A character-level tagging system is used in (Pennell and Liu, 2010) to solve deletion-based abbreviation. It was further extended in (Liu et al., 2012) using more characters instead of `Y` or `N` as labels. The character-level machine translation (MT) approach (Pennell and Liu, 2011) was modified in (Li and Liu, 2012a) into character-block. While a string edit distance method was introduced in (Contractor et al., 2010) to represent word-level similarity, and this orthographical feature has been adopted in (Han and Baldwin, 2011), and (Yang and Eisenstein, 2013).

Challenges are encountered in these different levels of tweet normalization. In the character-level sequential labeling systems, features are required for every character and their combinations, leading to much more noise into the later reverse table look-up process (Liu et al., 2012). In the character-block level MT systems equal number of blocks and their corresponding phonetic symbols are required for alignment (Li and Liu, 2012b). This strict restriction can result in a great difficulty in training set construction and a loss of useful information. Finally, word-level normalization methods cannot properly model how non-standard words are formed, and some patterns or consistencies within words can be omitted and altered.

We observe the cognitive process that, given non-standard words like `tmr`, people tend to first segment them into syllables like `t-m-r`. Then they will find the corresponding standard word with syllables like `to-mor-row`. Inspired by this cognitive observation, we propose a syllable based tweet normalization method, in which non-standard words are first segmented into syllables. Since we cannot predict the writers deterministic intention in using `tmr` as a segmentation of `tm-r`

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 920–928,
Beijing, China, July 26-31, 2015.

(representing tim-er) or t-m-r (representing to-mor-row), every possible segmentation form is considered. Then we represent similarity of standard syllables and non-standard *syllables* using an exponential potential function. After every transition probabilities of standard syllable and non-standard syllable are assigned, we then use noisy channel model and Viterbi decoder to search for the most possible standard candidate in each tweet sentence.

Our empirical study reveals that syllable is a proper level for tweet normalization. The syllable is similar to character-block but it represents phonetic features naturally because every word is pronounced with syllables. Our syllable-based tweet normalization method utilizes effective features of both character- and word-level: (1) Like character-level, it can capture more detailed information about how non-standard words are generated; (2) Similar to word-level, it reduces a large amount of noisy candidates. Instead of using domain-specific resources, our method makes good use of standard words to extract linguistic features. This makes our method extendable to new normalization tasks or domains.

The rest of this paper is organized as follows: previous work in tweet normalization are reviewed and discussed in Section 2. Our approach is presented in Section 3. In Section 4 and Section 5, we provide implementation details and results. Then we make some analysis of the results in Section 6. This work is finally concluded in Section 7.

2 Related Work

Non-standard words exhibit different forms and change rapidly, but people can still figure out their original standard words. To properly model this human ability, researchers are studying what remain unchanged under this dynamic characteristic. Human normalization of an non-standard word can be as follows: After realizing the word is non-standard, people usually first figure out standard candidate words in various manners. Then they replace the non-standard words with the standard candidates in the sentence to check whether the sentence can carry a meaning. If not, they switch to a different candidate until a good one is found. Most normalization methods in existence follow the same procedure: candidates are first generated, and then put into the sentence to check whether a reasonable sentence can be formed. Differences lie in how the candidates are generated and weighted. Related work can be classified into three groups.

2.1 Orthographical similarity

Orthographical similarity is built upon the assumption that the non-standard words *look like* its standard counterparts, leading to a high *Longest Common Sequence* (LCS) and low *Edit Distance* (ED). This method is widely used in spell checker, in which the LCS and ED scores are calculated for weighting possible candidates. However, problems are that the correct word cannot always be the most *looked like* one. Taking the non-standard word nite for example, note looks more likely than the correct form night. To overcome this problem, an *exception dictionary* of strongly-associated word pairs are constructed in (Gouws et al., 2011). Further, these pairs are added into a unified log-linear model in (Yang and Eisenstein, 2013) and Monte Carlo sampling techniques are used to estimate parameters.

2.2 Phonetic similarity

The assumption underlying the phonetic similarity is that during transition, non-standard words *sound like* the standard counterparts, thus the pronunciation of non-standard words can be traced back to a standard dictionary. The challenge is the algorithm to annotate pronunciation of the non-standard words. Double Metaphone algorithm (Philips, 2000) is used to decode pronunciation and then to represent phonetic similarity by edit distance of these transcripts (Han and Baldwin, 2011). IPA symbols are utilized in (Li and Liu, 2012b) to represent sound of words and then word alignment-based machine translation is applied to generate possible pronunciation of non-standard words. And also, phoneme is used in (Liu et al., 2012) as one kind of features to train their CRF model.

2.3 Contextual similarity

It is accepted that after standard words are transformed into non-standard words, the meaning of a sentence remains unchanged. So the normalized standard word must carry a meaning. Most researchers use n-gram language model to normalize a sentence, and several researches use more contextual information. For example, training pairs are generated in (Liu et al., 2012) by a cosine contextual similarity formula whose items are defined by TF-IDF scheme. A bipartite graph is constructed in (Hassan and Menezes, 2013) to represent tokens (both non-standard and standard words) and their context. Thus, random walks on the graph can represent contextual-similarity between non-standard and standard words. Very recently, word-embedding (Mikolov et al., 2010; Mikolov et al., 2013) is utilized in (Li and Liu, 2014) to represent more complex contextual relationship.

In word-to-word candidate selection, most researches use orthographical similarity and phonetic similarity separately. In the log-linear model (Yang and Eisenstein, 2013), edit distance is modeled as major feature. In the character- and phone-based approaches (Li and Liu, 2012b), orthographical information and phonetic information were treated separately to generate candidates.

In (Han and Baldwin, 2011), candidates from lexical edit distance and phonemic edit distance are merged together. Then an up to 16% increasing recall was reported when adding candidates from phonetic measure. But improper processing level makes it difficult to model the two types of information simultaneously: (1) Single character can hardly reflect orthographical features of one word. (2) As fine-grained reasonable restrictions are lacked, as showed in (Han and Baldwin, 2011), several times of candidates are included when adding phonetic candidates and this will bring much more noise. To combine orthographical and phonetic measure in a fine-grained level, we proposed the syllable-level approach.

3 Approach

3.1 Framework

The framework of the proposed tweet normalization method is presented in Figure 1. The proposed method extends the basic HMM channel model (Choudhury et al., 2007; Cook and Stevenson, 2009) into syllable level. And the following four characteristics are very intersting.

(1) **Combination**: When reading a sentence, fast subvocalization will occur in our mind. In the process, some non-standard words generated by phonetic substitution are correctly pronounced and then normalized. And also, because subvocalization is fast, people tend to ignore some minor flaws in spelling intentionally or unintentionally. As this often occurs in people's real-life interacting with these social media language, we believe the combination of phonetic and orthographical information is of great significance.

(2) **Syllable level**: Inspired by Chinese normalization (Xia et al., 2006) using pinyin (phonetic transcripts of Chinese), syllable can be seen as basic unit when processing pronunciation. Different from mono-syllable Chinese words, English words can be multi-syllable; this will bring changes in our method that extra layers of syllables must be put into consideration. Thus, apart from word-based noisy-channel model, we extend it into a syllable-level framework.

(3) **Priori knowledge**: Priori knowledge is acquired from standard words, meaning that both standard syllabification and pronunciation can shed some lights to non-standard words. This assumption makes it possible to obtain non-standard syllables by standard syllabification and gain pronunciation of syllables by standard words and rules generated with them.

(4) **General patterns**: Social media language changes rapidly while labeled data is expensive thus limited. To effectively solve the problem, linguistic features instead of statistical features should be emphasized. We exploit standard words of their syllables, pronunciation and possible transition patterns and proposed the four-layer HMM-based model (see Figure 1).

In our method, non-standard words c_i are first segmented into syllables $sc_i^{(1)} \dots sc_i^{(k)}$, and for standard syllable $sw_i^{(j)}$ mapping to non-standard syllable $sw_i^{(j)}$, we calculate their similarity by combining the orthographical and phonetic measures. Standard syllables $sw_i^{(1)} \dots sw_i^{(k)}$ make up one standard candidates. Since candidates are generated and weighted, we can use Viterbi decoder to perform sentence normalization. Table 1 shows some possible candidates for the non-standard word `tmr`.

3.2 Method

We extend the noisy channel model to syllable-level as follows:

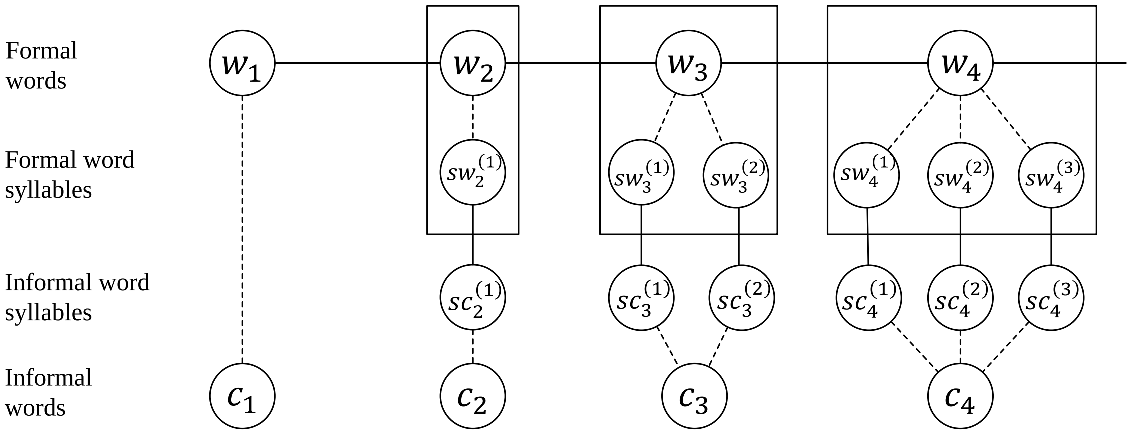

Figure 1: Framework of the propose tweet normalization method.

$$\begin{aligned}\widehat{w} &= argmax \quad p(w|c) \\ &= argmax \quad p(c|w) \times p(w) \\ &= argmax \quad p(\vec{sc}|\vec{sw}) \times p(\vec{sw}),\end{aligned} \tag{1}$$

where w indicates the standard word and c the non-standard word, and sw and sc represent their syllabic form, respectively. To simplify the problem, we restrict the number of standard syllables equals to the number of non-standard syllables in our method.

Assuming that syllables are independent of each other in transforming, we obtain:

$$p(\vec{sc}|\vec{sw}) = \prod_{j=1}^{k} p(sc_j|sw_j). \tag{2}$$

For syllable similarity, we use an exponential potential function to combine orthographical distance and phonetic distance. Because pronunciation can be represented using letter-to-phone transcripts, we can treat string similarity of these transcripts as phonetic similarity. Thus the syllable similarity can be calculated as follows.

tmr	t-mr	tm-r	t-m-r
tamer	ta-mer ti-mor ti-more tu-mor …	tim-er tim-ber ton-er tem-per …	to-mor-row tri-mes-ter tor-men-tor ta-ma-ra …

Table 1: Standard candidates of *tmr* in syllable level. The first row gives the different segmentations and the second row presents the candidates.

$$p(sc_j|sw_j, \lambda) = \frac{\Phi(sc_j, sw_j)}{Z(sw_j)} \tag{3}$$

$$Z(sw_j) = \sum_{sc_j} \Phi(sc_j, sw_j) \tag{4}$$

$$\begin{aligned}\Phi(sc, sw) = exp(\lambda(LCS(sc, sw) - ED(sc, sw)) \\ +(1-\lambda)(PLCS(sc, sw) - PED(sc, sw)))\end{aligned} \tag{5}$$

Exponential function grows tremendously as its argument increases, so much more weight can be assigned if syllables are more similar. The parameter λ here is used to empirically adjust relative contribution of letters and sounds. Longest common sequence (LCS) and edit distance (ED) are used to measure orthographical similarity, while phonetic longest common sequence (PLCS) and phonetic edit distant (PED) are used to measure phonetic similarity but based on letter-to-sound transcripts. The PLCS are defined as basic LCS but PED here is slightly different.

When performing phonetic similarity calculation based on syllables, we follow (Xia et al., 2006) in treating consonant and vowels separately because transition of consonants can make a totally different pronunciation. So if consonants of sc_j and sw_j are exactly the same or fit rules listed in Table 2, $PED(sc_j, sw_j)$ equals to edit

Description	Rules	Examples
1. -ng as suffix: g-dropping	-n/-ng	do-in/do-ing, go-in/go-ing, talk-in/talk-ing, mak-in/mak-ing
2. -ng as suffix: n-dropping	-g/-ng	tak-ig/tak-ing, likig/lik-ing
3. suffix: z/s equaling	-z/-s, -s/-z	jamz/james, plz/please
4. suffix: n/m equaling	-m/-n, -n/-m	in-portant/im-portant, get-tim/get-ting
5. suffix: t/d equaling	-t/-d, -d/-t	shid/shit, shult/should
6. suffix: t-dropping	-/-t	jus/just, wha/what, mus/must, ain/ain't
7. suffix: r-dropping	-/-r	holla/holler, t-m-r/tomorrow
8. prefix: th-/d- equaling	d-/th-, th-/d-	de/the, dat/that, dats/that's, dey/they

Table 2: The consonant rules.

distance of letter-to-phone transcripts, or it will be assigned infinity to indicate that their pronunciation are so different that this transition can seldom happen. For example, as consonantal transition between suffix `z` and `s` can always happen, PED(`plz`,`please`) equals string edit distance of their transcripts. But as consonatal transition of `f` and `d` is rare, phonetic distance of `fly` and `sky` is assigned infinity. Note the consonant rules in Table 2 are manually defined in our empirical study, which represent the most commonly used ones.

3.3 Parameter

Parameter in the proposed method is only the λ in Equation (5), which represents the relative contribution of orthographical similarity and phonetic similarity. Because the limited number of annotated corpus, we have to enumerate the parameter in {0, 0.1, 0.2, ..., 1} in the experiment to find the optimal setting.

4 Implementation

The method described in the previous section are implemented with the following details.

4.1 Preprocessing

Before performing normalization, we need to process several types of non-standard words:

- **Words containing numbers**: People usually substitute some kind of sounds with numbers like `4/four`, `2/two` and `8/eight` or numbers can be replacement of some letters like `1/i`, `4/a`. So we replace numbers with its words or characters and then use them to generate possible candidates.

- **Words with repeating letters**: As our method is syllable-based, repeating letters for sentiment expressing (like `cooool`, (Brody and Diakopoulos, 2011)) can cause syllabifying failure. For repeating letters, we reduce it to both two and one to generate candidate separately. Then the two lists are merged together to form the whole candidate list.

4.2 Letter-to-sound conversion

Syllable in this work refers to orthographic syllables. For example, we convert word `tomorrow` into `to-mor-row`. However, when comparing the syllable of a standard word and that of a non-standard word, sound (i.e., phones) of the syllables are considered. Thus letter-to-sound conversion tools are required.

Several TTS system can perform the task according to some linguistic rules, even for non-standard words. The Double Metaphone algorithm used in (Han and Baldwin, 2011) is one of them. But it uses consonants to encode a word, which gives less information than we need. In our method, we use freeTTS (Walker et al., 2002) with CMU lexicon[1] to transform words into APRAbet[2] symbols. For example, word `tomorrow` is transcribed to {`T-UW M-AA R-OW`} and `tmr` to {`T M R`}.

4.3 Dictionary preparation

- **Dictionary #1: In-vocabulary (IV) words**

  Following (Yang and Eisenstein, 2013), our set of IV words is also based on the GNU aspell dictionary (v0.60.6). Differently, we use a collection of 100 million tweets (roughly the same size of Edinburgh Twitter corpus) because the Edinburgh Twitter corpus is no longer available due to Twitter policies. The

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[2] http://en.wikipedia.org/wiki/Arpabet

final IV dictionary contains 51,948 standard words.

- **Dictionary #2: Syllables for the standard words**

  Following (Pennell and Liu, 2010), we use the online dictionary[3] to extract syllables for each standard words. We encountered same problem when accessing words with prefixes or suffixes, which are not syllabified in the same format as the base words on the website. To address the issue, we simply regard these prefixes and suffixes as syllables.

- **Dictionary #3: Pronunciation of the syllables**

  Using the CMU pronouncing dictionary (Weide, 1998) and dictionary 2, and knowing all possible APRAbet symbol for all consonant characters, we can program to capture every possible pronunciation of all syllables in the standard dictionary.

4.4 Automatic syllabification of non-standard words

Automatic syllabification of non-standard words is a supervised problem. A straightforward idea is to train a CRF model on manually labeled syllables of non-standard words. Unfortunately, such a corpus is not available and very expensive to produce.

We assume that both standard and non-standard forms follow the same syllable rules (i.e., the cognitive process). Thus we propose to train the CRF model on the corpus of syllables of standard words (which is easy to obtain) to construct an automatic annotation system based on CRF++ (Kudo, 2005). In this work, we extract syllables of standard words from Dictionary #2 as training set. Annotations follow (Pennell and Liu, 2010) to identify boundaries of syllables and in our work, CRF++ can suggest several candidate solutions, rather than an optimal segmentation solution for syllable segmentation of the non-standard words. In the HMM channel model, the candidate solutions are included as part of the search space.

4.5 Language model

Using Tweets from our corpus that contain no OOV words besides hashtags and username mentions (following (Han and Baldwin, 2011)), the Kneser-Ney smoothed tri-gram language model is estimated using SRILM toolkit (Stolcke, 2002). Note that punctuations, hashtags, and username mentions have some syntactic value (Kaufmann and Kalita, 2010) to some extent, we replace them with '<PUNCT>', '<TOPIC>' and '<USER>'.

5 Evaluation

5.1 Datasets

We use two labeled twitter datasets in existence to evaluate our tweet normalization method.

- **LexNorm1.1** contains 549 complete tweets with 1184 non-standard tokens (558 unique word type) (Han and Baldwin, 2011).

- **LexNorm1.2** is a revised version of LexNorm1.1 (Yang and Eisenstein, 2013). Some inconsistencies and errors in LexNorm1.1 are corrected and some more non-standard words are properly recovered.

In both datasets, to-be-normalized non-standard words are detected manually as well as the corresponding standard words.

5.2 Evaluation criteria

Here we use precision, recall and F-score to evaluate our method. As normalization methods on these datasets focused on the labeled non-standard words (Yang and Eisenstein, 2013), recall is the proportion of words requiring normalization which are normalized correctly; precision is the proportion of normalizations which are correct. When we perform the tweet normalization methods, every error is both a false positive and false negative, so in the task, precision equals to recall.

5.3 Sentence level normalization

We choose the following prior normalization methods:

- (Liu et al., 2012): the extended character-level CRF tagging system;

- (Yang and Eisenstein, 2013): log-linear model using string edit distance and longest common sequence measures as major features;

- (Hassan and Menezes, 2013): bipartite graph major exploit contextual similarity;

[3] http://www.dictionary.com

Method	Dataset	Precision	Recall	F-measure
(Han and Baldwin, 2011)	LexNorm 1.1	75.30	75.30	75.30
(Liu et al., 2012)		84.13	78.38	81.15
(Hassan and Menezes, 2013)		85.37	56.4	69.93
(Yang and Eisenstein, 2013)		82.09	82.09	82.09
Syllable-based method		85.30	85.30	85.30
(Yang and Eisenstein, 2013)	LexNorm 1.2	82.06	82.06	82.06
Syllable-based method		86.08	86.08	86.08

Table 3: Experiment results of the tweet normalization methods.

- (Han and Baldwin, 2011): the orthography-phone combined system using lexical edit distance and phonemic edit distance.

In our method, we set λ=0.7 because it is found best in our experiments (see Figure 2). The experimental results are presented in Table 3, which indicate that our method outperforms the state-of-the-art methods. Details on how to adjust parameter is given in Section 5.4.

Recall we argue that combination of three similarity is necessary when performing sentence-level normalization. Apart from contextual similarity like language model or graphic model, methods in (Yang and Eisenstein, 2013) or (Hassan and Menezes, 2013) do not include phonetic measure, causing loss of important phonetic information. Though using phoneme, morpheme boundary and syllable boundary as features (Liu et al., 2012), the character-level reversed approach will bring much more noise into the later reversed look-up table, and also, features of whole word are omitted.

Like (Han and Baldwin, 2011), we also use lexical measure and phonetic measure. Great difference between the two approaches is the processing level: word level and syllable level. In their work, average candidates number suffers times of increase when adding phonetic measure. This is because when introducing phonemic edit distance, important pronunciations can be altered (phonemic edit distance of `night-need` and `night-kite` is equal). Syllable level allows us to reflect consistencies during transition in a finer-grained level. Thus the phonetic similarity can be more precisely modeled.

5.4 Contributions of phone and orthography

In our method, the parameter λ in Equation 5 is used to represent the relatively contributions of both phonetic and orthographical information. But as the lack of prior knowledge, we cannot judge an optimal λ. We choose to conduct experiments varying $\lambda = \{0, 0.1, ..., 1\}$ to find out how this adjustment can affect performance. The experimental results are presented in Figure 2.

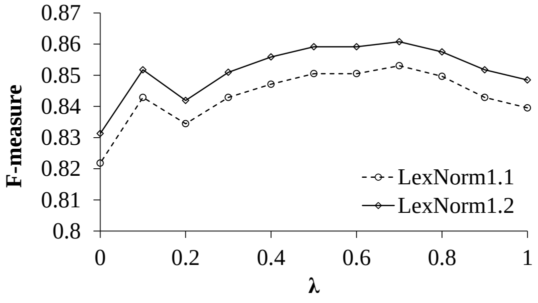

Figure 2: Contribution of phone and orthography.

As shown in Figure 2, when λ is set 0 or 1 (indicating no contribution of either orthographical or phonetic in assigning weight to candidates), our method performs much worse. In our experiment, when $\lambda = 0.7$, the models performs best, showing that orthographical measure makes relatively more contribution over phonetic measure, but the latter is indispensable. This justifies the effectiveness of combining orthographical and phonetic measure, indicating that human normalization process is properly modeled.

6 Analysis

6.1 Our exceptions

Deeper observation of our normalization results shows that there are several types of exceptions beyond our consonant-based rules. For example, `thanks` fails to be selected as a candidate for the non-standard word `thx` because the pronunciation of `thanks` contains an `N` but `thx` does not. The same situation happens when we process `stong/strong` because of the lacking `R`. We believe some more consonant should be exploited and more precisely described.

6.2 Non-standard words involving multiple syllables

There are one type of transition that we cannot solve like `acc/accelerate` and `bio/biology` because the mapping is between single-syllable word and multi-syllable word. We add possible standard syllable $sw_0^{(i)}$ and $sw_{k+1}^{(i)}$ to the head and tail of origin syllables, but this extended form failed to be assigned high probability because the string edit distances are too large. We leave this problem for further research.

6.3 Annotation issue

Though similar, our results of LexNorm1.2 is better than LexNorm1.1. After scrutinizing, we notice that several issues in LexNorm1.1 are fixed in LexNorm1.2. So our results like `meh/me` (meaning the non-standard word `meh` are corrected to `me`) in LexNorm1.1 is wrong but in LexNorm1.2 is right. Even in LexNorm1.2, there exist some inconsistencies and errors. For example, our result `buyed/bought` is wrong for both datasets, which is actually correct. For another example, `til` is normalized to `until` in some cases but to `till` in other cases. We show that the LexNorm test corpus is still imperfect. We appeal for systematic efforts to produce a standard dataset under a widely-accepted guideline.

6.4 Conventions

Social media language often contains words that are culture-specific and widely used in daily life. Some word like `congrats`, `tv` and `pic` are included into several dictionaries. We also observed several transitions like `atl/atlanta` or `wx/weather` in the datasets. These kinds of conventional abbreviations pose great difficulty to us. Normalization of those conventional non-standard words still needs further study.

7 Conclusion

In this paper, a syllable-based tweet normalization method is proposed for social media text normalization. Results on publicly available standard datasets justify our assumption that syllable plays a fundamental role in social media non-standard words. Advantage of our proposed method lies in that syllable is viewed as the basic processing unit and syllable-level similarity. This accords to the human cognition in creating and understanding the social non-standard words. Our method is domain independent. It is robust on non-standard words in any period of history. Furthermore, give the syllable transcription tool, our method can be easily adapted to a new language.

Acknowledgement

This research work was carried out when the authors worked at Tsinghua University. We acknowledge the financial support from Natural Science Foundation of China (NSFC: 61272233, 61373056, 61433018). We thank the anonymous reviewers for the insightful comments.

References

Samuel Brody and Nicholas Diakopoulos. 2011. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs. In *EMNLP*, pages 562–570. ACL.

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(3-4):157–174.

Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. 2010. Unsupervised cleansing of noisy text. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING (Posters)*, pages 189–196. Chinese Information Processing Society of China.

Paul Cook and Suzanne Stevenson. 2009. An unsupervised model for text message normalization. In *CALC '09: Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 71–78, Morristown, NJ, USA. Association for Computational Linguistics.

Stephan Gouws, Dirk Hovy, and Donald Metzler. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90. Association for Computational Linguistics.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *ACL*, pages 368–378. The Association for Computer Linguistics.

Hany Hassan and Arul Menezes. 2013. Social text normalization using contextual graph random walks. In *ACL (1)*, pages 1577–1586. The Association for Computer Linguistics.

Max Kaufmann and Jugal Kalita. 2010. Syntactic normalization of Twitter messages. In *International conference on natural language processing, Kharagpur, India*.

Taku Kudo. 2005. Crf++: Yet another crf toolkit. *Software available at http://crfpp. sourceforge. net*.

Chen Li and Yang Liu. 2012a. Improving text normalization using character-blocks based models and system combination. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 1587–1602.

Chen Li and Yang Liu. 2012b. Normalization of text messages using character- and phone-based machine translation approaches. In *INTERSPEECH*. ISCA.

Chen Li and Yang Liu. 2014. Improving text normalization via unsupervised model and discriminative reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Student Research Workshop*, pages 86–93.

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011. Insertion, deletion, or substitution?: normalizing text messages without precategorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 71–76. Association for Computational Linguistics.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *In Proceedings of ACL: Long Papers-Volume 1*, pages 1035–1044. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Luk Burget, Jan ernock, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, pages 1045–1048.

Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Deana Pennell and Yang Liu. 2010. Normalization of text messages for text-to-speech. In *ICASSP*, pages 4842–4845. IEEE.

Deana Pennell and Yang Liu. 2011. A character-level machine translation approach for normalization of sms abbreviations. In *IJCNLP*, pages 974–982.

S. Petrovic, M. Osborne, and V. Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT Workshop on Computational Linguistics in a World of Social Media*, pages 25–26.

Lawrence Philips. 2000. The double metaphone search algorithm. *C/C++ Users Journal*, 18(5), June.

Richard Sproat, Alan W. Black, Stanley F. Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November.

Willie Walker, Paul Lamere, and Philip Kwok. 2002. Freetts: a performance case study.

Robert L Weide. 1998. The cmu pronouncing dictionary. *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*.

Yunqing Xia, Kam-Fai Wong, and Wenjie Li. 2006. A phonetic-based approach to chinese chat text normalization. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, *ACL*. The Association for Computer Linguistics.

Yi Yang and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *EMNLP*, pages 61–72. ACL.