Toward Tweets Normalization Using Maximum Entropy

Mohammad Arshi Saloot
Department of Artificial Intelligence, University of Malaya, 50603, Malaysia
phd_siamak@yahoo.com

Norisma Idris
Department of Artificial Intelligence, University of Malaya, 50603, Malaysia
norisma@um.edu.my

Liyana Shuib
Department of Information System, University of Malaya, 50603, Malaysia
liyanashuib@um.edu.my

Ram Gopal Raj
Department of Artificial Intelligence, University of Malaya, 50603, Malaysia
ramdr@um.edu.my

AiTi Aw*
Institute for Infocomm Research (I2R), A*STAR, Singapore
aaiti@i2r.a-star.edu.sg

*Corresponding author

Abstract

The use of social network services and microblogs, such as Twitter, has created valuable text resources, which contain extremely noisy text. Twitter messages contain so much noise that it is difficult to use them in natural language processing tasks. This paper presents a new approach using the maximum entropy model for normalizing Tweets. The proposed approach addresses words that are unseen in the training phase. Although the maximum entropy needs a training dataset to adjust its parameters, the proposed approach can normalize unseen data in the training set. The principle of maximum entropy emphasizes incorporating the available features into a uniform model. First, we generate a set of normalized candidates for each out-of-vocabulary word based on lexical, phonemic, and morphophonemic similarities. Then, three different probability scores are calculated for each candidate using positional indexing, a dependency-based frequency feature and a language model. After the optimal values of the model parameters are obtained in a training phase, the model can calculate the final probability value for candidates. The approach achieved an 83.12 BLEU score in testing using 2,000 Tweets. Our experimental results show that the maximum entropy approach significantly outperforms previous well-known normalization approaches.

1 Introduction

The advent of Web 2.0 and electronic communications has enabled the extensive creation and dissemination of user-generated content (UGC). The UGC collections provide invaluable data sources in order to mine and extract beneficial information and knowledge, while, at the same time, resulting in less standardized language (Clark & Araki, 2011; Daugherty, Eastin, & Bright, 2008).

However, such content diverges from standard writing conventions. As shown by experts (Bieswanger, 2007; Thurlow & Brown, 2003), this divergence is due to the usage of a variety of coding strategies, including digit phonemes (*you too* → *you2*), phonetic transcriptions (*you* → *u*), vowel drops (*dinner* → *dnnr*), misspellings (*convenience* → *convineince*), and missing or incorrect punctuation marks (*If I were you, I'd probably go.* → *If I were you Id probably go*). These alterations are due to three main parameters: 1) The small allowance of characters, 2) the constraints of the small keypads, and 3) using UGC in informal communications between friends and relatives.

Whatever their causes, these alterations considerably affect any standard natural language processing (NLP) system, due to the presence of many out of vocabulary (OOV) words, also known as non-standard words (NSWs) and unknown words. Therefore, a text normalization process must be performed before any conven-

Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, pages 19–27,
Beijing, China, July 31, 2015. ©2015 Association for Computational Linguistics

tional NLP process is implemented (Sproat et al., 2001). As defined by Liu, Weng, Wang, and Liu (2011), "Text message normalization aims to replace the non-standard tokens that carry significant meanings with the context-appropriate standard words."

This paper proposes a novel normalization approach for Twitter messages. Twitter is the most popular microblogging service in the world for news-casting, sharing thoughts, and staying in touch with friends. Since its initial founding in 2006, it has gathered hundreds of millions of registered users. Tweets refer to messages sent on Twitter, which is restricted to 140 characters, 20 characters less than the 160 allowed by SMS. Because of this limitation, users have to transcribe Tweets with as much brevity as possible.

The normalization bears a resemblance to spelling correction. The ultimate goal of which is the detection and correction of OOV words. The spelling correction methods only focus on misspelled words while normalization systems consider all forms of OOV words, such as representing sounds phonetically (e.g. *by the way* → *btw*) and shortened forms (e.g. *university* → *uni*). Thus, normalization approaches should address a higher volume of OOV words compared to spelling correction approaches that lead to more complexity.

To address this complexity, we use maximum entropy (Berger, Pietra, & Pietra, 1996; Och & Ney, 2002) for utilizing and incorporating more probability functions. Our approach is based on the hypothesis that integrating more probability functions will boost the performance of the method; however, the available information and number of probability functions for (*OOV word, standard word*) pairs are always limited. Maximum entropy (Maxent) provides a criterion for integrating probability distributions based on partial knowledge. The Maxent produces the lowest biased estimation on the given information, that is, it is maximally neutral regarding missing information. When defining some unknown events with a statistical model, we should always select the one that has maximum entropy. Although the Maxent has already been used in the normalization sphere (e.g. Pennell and Liu (2010) utilized Maxent to classify deletion-based abbreviations), this paper explains how to employ Maxent for selecting the best-normalized candidate.

We have developed a method that does not require annotated training data and it normalizes unseen data. Most of the normalization approaches substantially depend on the manually annotated data, while the labeled data is costly and time consuming to prepare. We generate normalized candidates for each detected OOV based on lexical, phonemic, and morphophonemic variations. In addition, since our target dataset encompasses Twitter messages from Singaporeans and code-switching between Malay and English is frequent in the dataset, a Malay-English dictionary is utilized to generate candidates for Malay words. Finally, maximum entropy presents a backbone to combine several conditional probabilities of normalized candidates.

The remainder of this paper is organized as follows: Section 2 gives a survey of different approaches of normalizing noisy text. Section 3 describes the preprocessing stage. Section 4 illustrates the candidate generation stage. The proposed candidate selection method is demonstrated in Section 5. Finally, Section 6 concludes this paper with a summary and future works.

2 Related work

The normalization approaches can be categorized into four groups. The first group is called statistical machine translation (SMT) paradigm that addresses the normalization problem as a statistical machine translation task. This paradigm was first introduced by Aw, Zhang, Xiao and Su (2006) to normalize SMS text that translates a source language (UGC) to a target language (standard language). This paradigm has since been re-examined, expanded and improved by other researchers (Lopez Ludeña, San Segundo, Montero, Barra Chicote, & Lorenzo, 2012). For example, Kaufmann and Kalita (2010) used the SMT-like approach to normalize English Tweets.

To normalize SMS language, a supervised noisy channel model was introduced by Choudhury, Saraf, Jain, Sarkar, and Basu (2007) that used a hidden Markov model (HMM). This approach mimics the spell checking task that tries to handle the normalization problem via noisy channel models that study the UGC text as a noisy version of standard language. This paradigm has been scrutinized and enhanced by other researchers (Liu et al., 2011; Xue, Yin, & Davison, 2011a). For example, Cook and Stevenson (2009) modified this approach to design an unsupervised method using probabilistic models for only three common abbreviation types: stylistic variation, prefix clipping, and subsequence abbreviation. In addition, Beaufort, Roekhaut, Cougnon, and Fairon (2010) merged the SMT-like and the spell checking approaches to normalize French SMSs.

The third group is the dictionary based normalization approach, which is an easy-to-use and fast solution. This approach requires a dictionary whose entries are OOV and standard form pairs. It has been proven that using a colloquial dictionary can outperform some state-of-the-art and complex approaches (Clark & Araki, 2011; Saloot, Idris, & Mahmud, 2014). However, its performance highly relies on the size of the dictionary. Therefore, Han, Cook, and Baldwin (2012) introduced a method to automatically compile a large dictionary. To address the shortcomings of the dictionary approach, Oliva, Serrano, Del Castillo, and Igesias (2013) introduced a special Spanish phonetic dictionary, in which each entry is formed by a coded consonant string, vowels strings, and their positions in the word, for normalizing Spanish SMS texts.

The fourth group resembles automatic speech recognition (ASR) systems. This paradigm consists of three steps: 1) converting the text to strings of phonemes via letter-to-phone rules, 2) converting the strings of phonemes to words via pronunciation dictionaries, and 3) choosing the most probable words. The ASR-like approach has been merged with other approaches to boost its performance. Kobus, Yvon, and Damnati (2008) combined ASR-like and SMT-like approaches to normalize French SMSs. Lin, Bilmes, Vergyri, and Kirchhoff (2007) used this approach to detect OOV words in switchboard data.

Han and Baldwin (2011) illustrated a lexical method for normalizing Twitter messages. After detecting OOVs, ill-formed words, and generating a set of candidates, the best candidate is selected using a variety of metrics: lexical edit distance, phonemic edit distance, longest common subsequence (LCS), affix substring, language model, and dependency-based frequency features. The method achieved a 93.4 BLEU score in normalizing 549 English Tweets. This inspired us to design a normalization method that has three major stages: preprocessing, candidate generation, and candidate selection.

3 Preprocessing

First, we perform some initial text refining on the tweets. For example, consecutive whitespace characters are trimmed to single whitespace, and extra whitespaces are removed from the beginning and end of Tweets. The initial stage of most NLP tasks is the tokenization. Existing tokenization methods can perform accurately when the text is thoroughly clean, such as news feeds and book datasets. For example, the PTB-Tokenizer is a fast, deterministic, and efficient tokenization method. On the other hand, UGC text demands special methods due to irregularities in its whitespaces and punctuation. As suggested by Lopez Ludeña et al. (2012), we employ a straightforward word separating method, which performs tokenization based on whitespace characters.

One of the most important primary steps in unsupervised normalization systems is to detect OOV words. Hanspell and GNU Aspell are two well-known spell checker systems, however, Aspell performance is more accurate on the noisy text (Clark & Araki, 2011). The Aspell dictionary is utilized to distinguish between OOV and standard English words. In addition, we used seven regular expression rules, which were introduced by Saloot, Idris, and Aw (2014). This helps to detect proper nouns, email and URL addresses, Twitter special symbols, and digits. The potential errors in the OOV word detection step would not affect the performance of the normalization system since the detected OOV word will be included in the candidate set.

4 Candidate generation

For each given OOV word, a set of normalized candidates is generated via four different modules. The first module executes a lexical candidate generation, which is extensively utilized in spell checker systems. It calculates candidates within a distance of T edit operations of the detected OOV words. Han and Baldwin (2011) stated that when T is less than or equal to two, the level of recall is high enough. The edit distance is the number of applied edits in changing one word to another. An edit could be a deletion, transposition, alteration, or insertion. Studies in spelling correction found that one lexical edit distance covers 80% to 95% of errors, and two lexical edit distances cover 98% of them. Therefore, here we use lexical variations with less than or equal to two edit distances.

For a word of length n characters, $54n + 25$ combinations will be generated with one lexical edit distance using four reshaping strategies: 1) Deletion strategy eliminates characters in all possible positions (e.g. *aer* → *er*, *ar*, *ae*), which generates n combinations. 2) Transposition strategy switches two adjacent characters (e.g. aer →

ear, *are*), which generates $n-1$ combinations. 3) Alteration strategy substitutes each character with all English alphabets (e.g. *aer* → *ber*, *cer*, *der*, *eer*, *fer*, *ger*, *her*, etc.), which generates $26n$ combinations. 4) Insertion strategy presumes that a letter is dropped, thus adding all the alphabets between characters (e.g. *aer* → *aaer*, *baer*, *caer*, *daer*, *eaer*, *faer*, *gaer*, *haer*, etc.), which generates $26(n+1)$ combinations. Finally, from the achieved combinations, standard words will be selected using the Aspell dictionary. However, many OOV words in Twitter are quite far from their target in term of edit distance especially in terms of deletions and substitutions. Therefore, we generated more candidates via three other methods.

Similar to the speech recognition systems, the second module generates candidates based on phoneme sounds. First, grapheme to phoneme conversion is performed using the Phonetisaurus tool (Novak, Yang, Minematsu, & Hirose, 2011). Phonetisaurus is an open-source phonetizer that is designed in the form of a weighted finite state transducer (WFST). After selecting the 10 best phoneme sequences, it looks up the phonemes in a pronouncing dictionary – Carnegie Mellon University (CMU) dictionary. The CMU is a machine-readable pronunciation dictionary that contains over 134,000 words including OOV words such as proper nouns and acronyms. Due to the existence of a large number of OOV words in the CMU dictionary, we filter out the OOVs using the Aspell dictionary.

The third module, as proposed by Saloot, Idris, and Aw (2014), is a combination of the two previous modules. First, it lexically generates candidates within one edit distance of the given OOV word, and then sends the candidates to the phoneme module. Since our testing dataset consists of English Tweets posted by Singaporeans, code-switching between Malay and English is frequent in the text. Therefore, our last module translates OOV words to English (if any). We searched for the tokens in the Smith Malay-English Dictionary (Smith & Padi, 2006), and inserted the meanings in the candidate set.

Table 1 displays the average number of generated candidates for each module. The lowest rate is associated with the Malay dictionary module. Two lexical edit operations generate the highest number of candidates, which indicates the highest recall and lowest precision. The rank of combination and phoneme modules are second and third, respectively.

No.	*module*	*Average number of candidates*
1.	Two lexical edit distance	70
2.	Combination	50
3.	Phoneme	20
4.	Malay dictionary	3

Table 1: The average number of generated candidates for five letter words.

5 Candidate selection

The main contribution of this work is to present a novel candidate selection method. The candidate selection stage consists of two steps: 1) assigning a variety of probability scores to candidates, and 2) integrating probability scores to select the best candidate. Our candidate selection method requires a training dataset. The training and testing datasets are collected from an extensive English Twitter corpus posted by Singaporeans (Saloot, Idris, Aw, & Thorleuchter, 2014). Three linguistic experts manually normalized 7,000 Tweets, while using inter-normalization agreement as an indicator. The experts were instructed to produce a text that is as close to standard English as possible, but leaves the Twitter special symbols (e.g. #topic and @username) as is. The dataset was split into two parts: 5,000 messages for the training phase, and 2,000 messages for the testing phase.

5.1 Calculation of probability scores

In order to select the most suitable candidates, we calculate their conditional probability scores using, positional indexing, a dependency-based frequency feature, and a language model (LM).

Inspired by work on a normalization dictionary (Han et al., 2012), the first method to calculate the probability score of the candidates is the positional indexing, which is widely used in information retrieval systems. The positional indexing deals with positional locations of term occurrences inside documents. To compile a positional index dataset, a method illustrated in Manning and Raghavan (2009) is applied on a cleansed portion of our Twitter corpus. Table 2 refers to an example of our achieved positional index dataset. Each Twitter message is considered as a single document, and, hence, a unique document ID is assigned to each document. The frequency value indicates the total number of appearances of a word in a document. The position values express the locations of the word in the document.

Vocab	*Document ID.*	*Frequency*	*Position*
have	1	2	4,9
	4	3	5, 11, 18
are	5	1	2
	12	2	2, 9
	14	2	2, 11

Table 2: An example of the positional indexes obtained.

A probability score is assigned to the normalized candidate according to a comparison between the position of the candidate and positional indexes in the dataset. We look for the candidate in the dataset where there is an occurrence of the candidate with its position index. After aggregating the number of occurrences, we normalize it between 0.0 and 1.0.

The next probability calculation method is the dependency-based frequency, which is an augmentation of the previous method. Inspired by a work on the lexical normalization of Tweets (Han & Baldwin, 2011), the noisy portion of our training dataset is parsed to obtain a dependency bank using our adapted version of the Stanford dependency parser (Marneffe, MacCartney, & Manning, 2006). Since our aim is not to perform actual dependency parsing, the dependency types are not extracted. A cleansed corpus is not utilized because the percentage of IV words is high enough in the corpus, and in the probability-measuring phase, OOV words are already detected. For example, from a sentence such as “*I will go to London by next week*,” (*next*, *go* +3) is obtained, indicating that *next* appears two words after *go*. The aggregations of all the dependency scores, which are called confidence scores, are stored in the dependency bank. A five-gram dependency bank is prepared without using a root node (head-word), that is, the process is iterated for all words in the sentence.

A probability score between 0.0 and 1.0 is assigned to each candidate. A relative position score in the form of (*candidate word*, *context word*, position) is calculated for each candidate within a context window of two words on either side. The obtained relative position of a candidate is compared with the existing confidence score in the dependency bank.

The third method of probability measurement calculates the probabilities based on a language model. The cleansed part of our training dataset, which consists of more than 55,000 words, is fed into SRILM (Stolcke, 2002) to compile a bidirectional trigram LM by employing the Kneser-Ney smoothing algorithm. To calculate the probability of each candidate, we used a beam search decoder through the Moses decoder (Koehn et al., 2007).

5.2 Selecting the most probable candidate

Previous works on spelling correction and normalization used the source channel model, which is also known as the noisy channel model and Naïve Bayes (Beaufort et al., 2010; Kernighan, Church, & Gale, 1990; Mays, Damerau, & Mercer, 1991; Toutanova & Moore, 2002; Xue, Yin, & Davison, 2011b). In the noisy channel approach, we observe the conversion of standard words to noisy words in a training phase in order to build a model. In the prediction phase, the decoder can select the most probable candidate based on the obtained model. The candidate selection is accomplished based on only two parameters: the LM and error model, which is computed as follows:

$$\begin{aligned} G &= \arg\max\left\{P(T \mid O)\right\} \\ &= \arg\max\left\{\sum_{m=1}^{M} \lambda_m \bullet f_m(T,O)\right\} \end{aligned}$$

Where T is a target word, O is an observed word, $f_m(T,O)$ is a feature function, M is a number of total feature functions, and λ is a Lagrange multiplier of each function. In our case, M equals three, in which f_1 is the positional indexing, f_2 is the dependency-based frequency feature, and f_3 is the LM probability. The Maxent requires λ being determined in the training phase before the actual usage.

6 Experimental results and discussion

We evaluate our approach in terms of BLEU score (Papineni, Roukos, Ward, & Zhu, 2002), since BLEU has become a well-known and adequate evaluation metric in normalization studies (Contractor, Faruquie, & Subramaniam, 2010; Schlippe, Zhu, Gebhardt, & Schultz, 2010). The achieved baseline for the testing dataset is 42.01 BLEU score, that is, the volume of similarity between the testing text and the reference text (manually normalized text) in term of BLEU score.

In the training phase, we performed maximum likelihood training (Papineni, Roukos, & Ward, 1998; Streit & Luginbuhl, 1994) for λ_1, λ_2 and λ_3 between 0.0 and 1.0. Figure 1 shows the tolerance of the performance while transition of λ_1 and λ_2 (when λ_3 is fixed to 1.0). Figure 1 depicts that the value of performance achieves the highest when the λ_1 and λ_2 are close to 0.63 and 0.9, respectively. It is found that the best performance is achieved by 0.6, 0.9, and 1.0 values for λ_1, λ_2, and λ_3, respectively. This means that LM has the highest impact on the candidate selection, and that dependency-based frequency has a higher impact on candidate selection than positional.

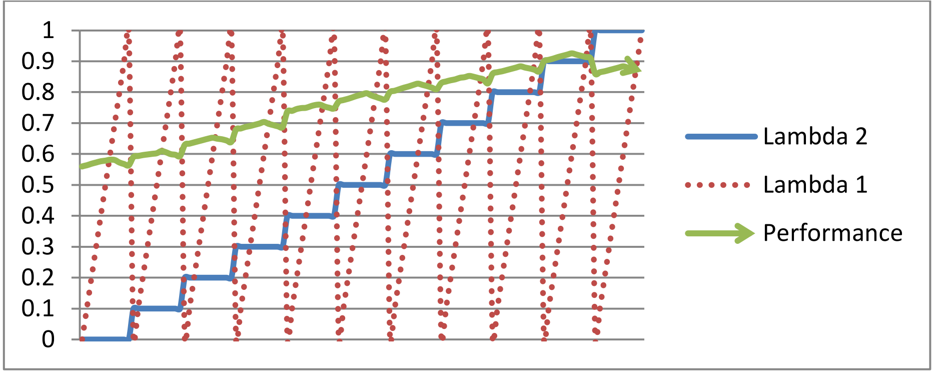

Figure 1: The training of Maxent for lambda settings.

We divided our dataset into six equal sets in order to perform 6-fold cross validation. As shown in Table 3, the average of the obtained BLEU scores in six evaluation rounds was 83.12. The evaluation proves that our approach boosts the BLEU score by 41.11 (i.e. from 42.01 to 83.12). Since previous normalization studies used different data sources in their experiments, a direct comparison between our accuracy values is not meaningful. Therefore, we re-examined one of the state-of-the-art approaches using our dataset.

6-fold cross validation	*BLEU score*
Round 1	80.99
Round 2	81.57
Round 3	84.82
Round 4	83.91
Round 5	83.90
Round 6	83.55
Average	83.12

Table 3: Normalization results for 6-fold cross validation test.

The statistical machine translation (SMT) is a cutting-edge approach that handles the normalization problem as a statistical machine translation task; it was first introduced by Aw, Zhang, Xiao, and Su (2006). The SMT-like approach translates a source language (UGC) to a target language (standard language). The experiment was performed using Moses (Koehn et al., 2007) for statistical translation, Giza++ (Och & Ney, 2003) for word alignment, and SRILM (Stolcke, 2002) for LM compiling. The SMT system is trained using our Twitter aligned dataset. The optimum results were achieved using a trigram LM and Backoff smoothing (Jelinek, 1990): 78.81 BLEU score.

Table 4 indicates some statistics about our testing dataset. The OOV words are those detected by our OOV detection module. The BLEU score of raw text is an important measure to analyze the difficulty of the task. It is important to note that the dataset used in our experiment contains an above average number of OOV words compared to the datasets in other related papers. The dataset used by Kobus et al. (2008) consists of 32% OOV words, which is slightly lower than 34% of our dataset. In addition, Aw et al. (2006) used a dataset with a baseline BLEU score of 57.84, which indicates that the raw text is much more similar to the manual translated text (reference text) than the ones used in our experiment.

Avg. length of words (character)	5
Avg. number of words	11
Total No. of tokens	19,759
OOV words	34.02%
BLEU score of raw text	42.01

Table 4: Statistics of testing dataset.

As shown in Table 4, the average length of words is five characters, which makes the normalization task more difficult. For example, the candidate set for the OOV word "*yoor*" contains 59 words, as shown in Table 5. The large number of candidates causes difficulty for candidate selection because more options lead to more possibilities and more computational cost. Furthermore, the generated candidates are lexically, syntactically, and semantically very akin to each other. For example, for the OOV word "*yoor*", "*our*" might be mistakenly selected instead of "*your*". There are a smaller number of potential candidates for lengthy OOV words. As shown in Table 5, the number of candidates for the OOV word "*acessibility*" is only 14, which is less than average, thereby making candidate selection easier. Moreover, there is a distinct difference between the meanings of candidates, which is an easy situation for our context-based probability functions to select the correct one. Although our approach obtained promising results on this dataset, it works better on long words.

OOV word	*Candidate set*	*No. of candidates*
acessibility	accessibility, accessibly, basicity, bicyclists, bicyclist, italicizes, abilities, bicyclist, sibilates, stabilize, silicates, celibacy, bicycles, and bicycle.	14
yoor	your, you, door, our, or, yoga, yak, yuck, yule, moon, tour, poor, …	59

Table 5: Example of candidate sets for OOV words.

Our approach and SMT-like system attained BLEU scores of 83.12 and 78.81, respectively. This result proves that if we integrate three probability scores via Maxent, promising normalization accuracy can be obtained. This result confirms that a normalization system constructed based on the Maxent principle can surpass state-of-the-art systems. However, several drawbacks of our method were disclosed by inspecting the output of the system. The most noticeable one is that the approach fails when tackling very noisy text, that is, ample usage of OOV words in a text. We altered our dataset to have higher levels of noise using an approach introduced by Gadde, Goutam, Shah, Bayyarapu, and Subramaniam (2011), which artificially generates OOV words. If the percentage of OOV words crosses 45%, the accuracy of the method drastically drops to a BLEU score of less than 65. Another shortcoming of our approach is that it is not able to address combined words and abbreviations (e.g. *alot* → *a lot*, *btw* → *by the way*) because candidate generation module forms only single words for each OOV.

7 Conclusion

In this paper, we have presented a normalization approach based on the maximum entropy model. This approach provides a unified layout for incorporating different sources of features to normalize Twitter messages. Our proposed approach consists of three stages: preprocessing, candidate generation, and candidate selection. The approach is robust to normalize unseen words since its candidate generation stage does not practice machine-learning methods. In the preprocessing stage, after trimming erroneous whitespaces and tokenization, OOV words are detected via the GNU Aspell dictionary. Normalized candidates are generated for each OOV word in the second stage regarding to lexical, phonemic, and morphophonemic similarities. Since code-switching between Malay and English is very common in our dataset, the potential English translation of OOV words is also added to the candidate set.

In the third stage, three conditional probability scores are assigned to each candidate: 1) positional indexing considers the probability of positional locations of term occurrences inside documents, 2) dependency-based frequency measures the probability of prevalence of the dependency relation of words to each other, and 3) the language model indicates the probability of distribution of the sequence of words. Finally, the best candidate is selected. Maximum entropy integrates the obtained probability scores to estimate the ultimate probability of each candidate.

The approach is examined using 7,000 parallel Twitter messages, which is split into 5,000 messages for training and 2,000 for testing. The result is promising whereby we achieve a BLEU score of 83.12 against the baseline BLEU, which scores 42.01. We have compared our approach with a SMT-like approach using the same dataset. The accuracy of the SMT-like was lower than our approach (i.e. 78.81 BLEU score for the SMT-like). For future work, we will examine the Maxent normalization approach with more probability functions, such as distributional clustering and semantic features.

Acknowledgments

The research for this paper was financially supported by the University of Malaya FRGS Grant (FP021-2014B). We thank Asad Abdi for assistance with graphical illustrations.

Reference

Aw, A., Zhang, M., Xiao, J., & Su, J. (2006). A Phrase-based Statistical Model for SMS Text Normalization. In Proceedings of the COLING/ACL on Main Conference Poster Sessions (pp. 33–40). Stroudsburg, PA, USA: Association for Computational Linguistics.

Beaufort, R., Roekhaut, S., Cougnon, L.-A., & Fairon, C. (2010). A Hybrid Rule/Model-based Finite-state Framework for Normalizing SMS Messages. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (pp. 770–779). Stroudsburg, PA, USA: Association for Computational Linguistics.

Berger, A. L., Pietra, V. J. Della, & Pietra, S. A. Della. (1996). A Maximum Entropy Approach to Natural Language Processing. Comput. Linguist., 22(1), 39–71.

Bieswanger, M. (2007). 2 abbrevi8 or not 2 abbrevi8: A Contrastive Analysis of Different Space- and Time-Saving Strategies in English and German Text Messages. Texas Linguistic Forum, Vol. 50.

Choudhury, M., Saraf, R., Jain, V., Sarkar, S., & Basu, A. (2007). Investigation and Modeling of the Structure of Texting Language, 63–70.

Clark, E., & Araki, K. (2011). Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English. Procedia - Social and Behavioral Sciences, 27(0), 2–11.

Contractor, D., Faruquie, T. A., & Subramaniam, L. V. (2010). Unsupervised Cleansing of Noisy Text. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 189–196). Stroudsburg, PA, USA: Association for Computational Linguistics.

Cook, P., & Stevenson, S. (2009). An Unsupervised Model for Text Message Normalization. In Proceedings of the Workshop on Computational Approaches to Linguistic Creativity (pp. 71–78). Stroudsburg, PA, USA: Association for Computational Linguistics.

Daugherty, T., Eastin, M. S., & Bright, L. (2008). Exploring Consumer Motivations for Creating User-Generated Content. Journal of Interactive Advertising, 8(2).

Gadde, P., Goutam, R., Shah, R., Bayyarapu, H. S., & Subramaniam, L. V. (2011). Experiments with Artificially Generated Noise for Cleansing Noisy Text. In Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data (pp. 4:1–4:8). New York, NY, USA: ACM. doi:10.1145/2034617.2034622

Han, B., & Baldwin, T. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (pp. 368–378). Stroudsburg, PA, USA: Association for Computational Linguistics.

Han, B., Cook, P., & Baldwin, T. (2012). Automatically Constructing a Normalisation Dictionary for Microblogs. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 421–432). Stroudsburg, PA, USA: Association for Computational Linguistics.

Jelinek, F. (1990). Readings in Speech Recognition. In A. Waibel & K.-F. Lee (Eds.), (pp. 450–506). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kaufmann, M., & Kalita, J. (2010). Syntactic normalization of Twitter messages. International Conference on Natural Language Processing, Kharagpur, India.

Kernighan, M. D., Church, K. W., & Gale, W. A. (1990). A Spelling Correction Program Based on a Noisy Channel Model. In Proceedings of the 13th Conference on Computational Linguistics - Volume 2 (pp. 205–210). Stroudsburg, PA, USA: Association for Computational Linguistics.

Kobus, C., Yvon, F., & Damnati, G. (2008). Normalizing SMS: Are Two Metaphors Better Than One? In Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1 (pp. 441–448). Stroudsburg, PA, USA: Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., … Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (pp. 177–180). Stroudsburg, PA, USA: Association for Computational Linguistics.

Lin, H., Bilmes, J., Vergyri, D., & Kirchhoff, K. (2007). OOV detection by joint word/phone lattice alignment. In Automatic Speech Recognition Understanding, 2007. ASRU. IEEE Workshop on (pp. 478–483). doi:10.1109/ASRU.2007.4430159

Liu, F., Weng, F., Wang, B., & Liu, Y. (2011). Insertion, Deletion, or Substitution?: Normalizing Text Messages Without Pre-categorization nor Supervision. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2 (pp. 71–76). Stroudsburg, PA, USA: Association for Computational Linguistics.

Lopez Ludeña, V., San Segundo, R., Montero, J. M., Barra Chicote, R., & Lorenzo, J. (2012). Architecture for Text Normalization using Statistical Machine Translation techniques. In IberSPEECH 2012 (pp. 112–122). Madrid, Spain: Springer.

Manning, C. D., & Raghavan, P. (2009). An Introduction to Information Retrieval. Online. doi:10.1109/LPT.2009.2020494

Marneffe, M.-C. de, MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In The International Conference on Language Resources and Evaluation (LREC) (pp. 449–454). Genova, Italy.

Mays, E., Damerau, F. J., & Mercer, R. L. (1991). Context based spelling correction. Information Processing & Management, 27(5), 517–522.

Novak, J., Yang, D., Minematsu, N., & Hirose, K. (2011). Phonetisaurus: A wfst-driven phoneticizer. The University of Tokyo, Tokyo Institute of Technology. Retrieved January 1, 2014, from http://code.google.com/p/phonetisaurus/

Och, F. J., & Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 295–302). Stroudsburg, PA, USA: Association for Computational Linguistics.

Och, F. J., & Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. Comput. Linguist., 29(1), 19–51.

Oliva, J., Serrano, J. I., Del Castillo, M. D., & Igesias, Á. (2013). A SMS Normalization System Integrating Multiple Grammatical Resources. Natural Language Engineering, 19(01), 121–141.

Papineni, K., Roukos, S., & Ward, T. (1998). Maximum likelihood and discriminative training of direct translation models. In Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on (Vol. 1, pp. 189–192 vol.1).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 311–318). Stroudsburg, PA, USA: Association for Computational Linguistics.

Pennell, D. L., & Liu, Y. (2010). Normalization of text messages for text-to-speech. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.

Saloot, M. A., Idris, N., & Aw, A. (2014). Noisy Text Normalization Using an Enhanced Language Model. In Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition (pp. 111–122). Kuala Lumpur, Malaysia: SDIWC.

Saloot, M. A., Idris, N., Aw, A., & Thorleuchter, D. (2014). Twitter corpus creation: The case of a Malay Chat-style-text Corpus (MCC). Digital Scholarship in the Humanities. Retrieved from http://dsh.oxfordjournals.org/content/early/2014/12/13/llc.fqu066.abstract

Saloot, M. A., Idris, N., & Mahmud, R. (2014). An architecture for Malay Tweet normalization. Information Processing & Management, 50(5), 621–633.

Schlippe, T., Zhu, C., Gebhardt, J., & Schultz, T. (2010). Text normalization based on statistical machine translation and internet user support. In T. Kobayashi, K. Hirose, & S. Nakamura (Eds.), INTERSPEECH (pp. 1816–1819). ISCA.

Smith, J., & Padi, P. (2006). Lets make a dictionary. In Proceedings of the the Eighth Biennial Conference of the Borneo Research Council (BRC) (pp. 515–520). Sarawak, Malaysia: Borneo Research Council (BRC).

Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., & Richards, C. (2001). Normalization of non-standard words. Computer Speech & Language, 15(3), 287–333.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In Proceedings International Conference on Spoken Language Processing (pp. 257–286).

Streit, R. L., & Luginbuhl, T. E. (1994). Maximum likelihood training of probabilistic neural networks. Neural Networks, IEEE Transactions on, 5(5), 764–783. doi:10.1109/72.317728

Thurlow, C., & Brown, A. (2003). Generation Txt? The sociolinguistics of young people’s text-messaging.

Toutanova, K., & Moore, R. C. (2002). Pronunciation Modeling for Improved Spelling Correction. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 144–151). Stroudsburg, PA, USA: Association for Computational Linguistics.

Xue, Z., Yin, D., & Davison, B. D. (2011a). Normalizing Microtext. In Analyzing Microtext (Vol. WS-11–05). AAAI.

Xue, Z., Yin, D., & Davison, B. D. (2011b). Normalizing Microtext. In Analyzing Microtext: Papers from the 2011 AAAI Workshop (pp. 74–79). San Francisco, CA, USA: AAAI.