Normalization of Text Messages Using Character- and Phone-based Machine Translation Approaches

Chen Li, Yang Liu

Computer Science Department, The University of Texas at Dallas, Richardson, TX, USA

{chenli,yangl}@hlt.utdallas.edu

Abstract

There are many abbreviation and non-standard words in SMS and Twitter messages. They are problematic for text-to-speech (TTS) or language processing techniques for these data. A character-based machine translation (MT) approach was previously used for normalization of non-standard words. In this paper, we propose a two-stage translation method to leverage phonetic information, where non-standard words are first translated to possible pronunciations, which are then translated to standard words. We further combine it with the single-step character-based translation module. Our experiments show that our proposed method significantly outperforms previous results in both n-best coverage and 1-best accuracy.

Index Terms: text normalization, text-to-speech, abbreviation

1. Introduction

Text messaging or SMS has been one of the most popular communication forms for many years. According to a half-year research [1] done by CTIA last year, American people sent 196.9 billion text messages in 2011 compared to 12.5 billion in 2006. Another statistics [2] showed that for the second quarter of 2008, U.S. mobile subscribers sent and received on average 357 text messages per month, compared with making and receiving 204 phone calls a month. Abbreviation or non-standard words are very common in text messages for various reasons (e.g., length limitation, need to convey much information, writing style). They post problems to text-to-speech (TTS) systems or automatic language processing techniques for this domain. For example, TTS systems might not correctly determine the pronunciation for the token '2mr' (meaning 'tomorrow').

Text normalization aims to convert the non-standard words to the corresponding standard words. This is usually the first and also a vital task for either TTS or natural language processing. However, this is not a trivial task because of the large variation of the non-standard words. Letters in a standard word can be deleted, replaced by a digit or punctuation, or even reordered to create a non-standard word.

Different methods have been used for text normalization in both formal domains such as news wire text and informal genre such as text messages. In this paper, we propose an enhanced normalization system based on the character-based machine translation (MT) model introduced by Pennell and Liu in [3]. Rather than directly translating non-standard words to standard words using surface character information, we use a two-stage approach where we first translate the non-standard words to possible phonetic forms, and then convert these phonetic forms to in-vocabulary words. We further combine this method with the previous single-step character-based machine translation method. Our experiments on a commonly used test set show that the combined system can both improve the top one precision and increase the N-best coverage.

This paper is organized as follows. Section 2 introduces some previous work in this area. Section 3 presents our approach. Our experiments are described in Section 4. At last, Section 5 gives conclusions and future work.

2. Related Work

Text normalization has been well studied for TTS. See [4] for a good report of this problem. Recently much work has been done to address the normalization problems for abbreviations or non-standard words, especially for informal domains such as text messages and blogs. A simple approach for normalization would be applying traditional spell checking model, which is usually based on edit distance [5], [6]. However, spell-checking algorithms do not always work well for non-standard words that are drastically different from their standard forms, such as 'ate' and 'eight'. Therefore, some prior work [7] combined edit distance with pronunciation models to handle this problem.

Another line of work in normalization adopts a noisy channel model. Instead of directly computing the probability of a standard word given a non-standard word, this method uses Bayes rule and evaluates the probability of a non-standard word given a standard word. In the noisy channel framework, the most possible standard word S for a non-standard word A is:

$$S = \arg\max p(S \mid A) = \arg\max p(A \mid S)p(S) \quad (1)$$

Different methods have been used to compute $p(A \mid S)$. [8] used HMM for SMS normalization. [9] extended the work by introducing an unsupervised training method. [10] used a CRF sequence modeling approach for deletion-based abbreviation. [11] extended Pennell's work ([10]) by allowing other abbreviation types and also using automatically collected training data. [12] used the noisy channel model to handle the normalization of Chinese chat language based on specific properties of Chinese. They measured the similarity of two Chinese characters by computing the initial pinyin (shengmu) and the final pinyin (yunmu). [13] modeled $p(S \mid A)$ by computing the grapheme and phoneme similarity and then combined those results with context channel and acronym channel.

Machine translation (MT) is another commonly chosen method for text normalization. [14] viewed SMS as another language with its own words and rules, then MT techniques were used to translate this sort of 'foreign language' to regular English. [15] tackled normalization through an ASR-like system based on the fact that text messages can be represented by phonetic symbols. An MT model was also used in [16], but the

focus of that work is to generate more non-standard and standard words to address the problem of the lack of training data. [3] firstly introduced a character-based MT method for normalization, which is the basis of our work in this paper. This character-level translation system is another way to solve $p(A \mid S)$ in Eq (1).

For the normalization task for sentences (or messages), a system needs to first identify words that need to be normalized. This is typically done by simply checking whether a word is in a given dictionary. [17] developed a model to determine whether an out-of-vocabulary (OOV) word is a non-standard word that needs normalization or it is just a well-formed OOV word. Then for those ill-formed OOV words, they used grapheme and phoneme level similarity to generate candidate words.

3. Approach

Our method is an extension of the character-based machine translation approach [3]. Instead of directly translating the character sequence, we propose a two-stage approach to leverage phonetic information, where we first translate the non-standard words to possible phonetic sequences, and then to proper words. We further combine this with the character-based translation module to take advantage of their complementary strengths. Figure 1 shows the framework of our method. The following explains each component in more details.

3.1. Single-step Character-based Translation

System A in Figure 1 is the character-based machine translation method introduced in [3]. Similar to machine translation for a word sequence, this method aims to translate the character sequence as seen in a non-standard word.

Formally, for a non-standard word A= $a_1 a_2 a_3 \ldots a_n$, the task is to find the most likely standard word S= $s_1 s_2 s_3 \ldots s_n$, where a_i and s_i are the characters in the words:

$$\begin{aligned} S &= \arg\max p(S \mid A) \\ &= \arg\max p(A \mid S) p(S) \\ &= \arg\max p(a_1 a_2 a_3 \ldots a_n \mid s_1 s_2 s_3 \ldots s_n) p(s_1 s_2 s_3 \ldots s_n) \end{aligned} \quad (2)$$

where $p(a_1 a_2 a_3 \ldots a_n \mid s_1 s_2 s_3 \ldots s_n)$ is from the machine translation model and $p(s_1 s_2 s_3 \ldots s_n)$ is from a character-level language model. The translation model is trained using a parallel corpus containing pairs of non-standard words and standard words, and the character n-gram language model can be trained using an English dictionary.

During testing, the translation module generates hypotheses of character sequences for a given non-standard word. We use an English dictionary to remove candidates that are not in the dictionary and preserve N-best candidates.

This character-based translation method has advantages over the word-based approach used in some previous work [14] in that it can translate non-standard words not seen in the training set since it learns translation patterns based on character sequences rather than the entire words. Note that non-alphabetic symbols are allowed in this model. For example, if training pairs contain '@' and 'at', 'every1' and 'everyone', then the model can learn mappings '@' and 'at', '1' and 'one'.

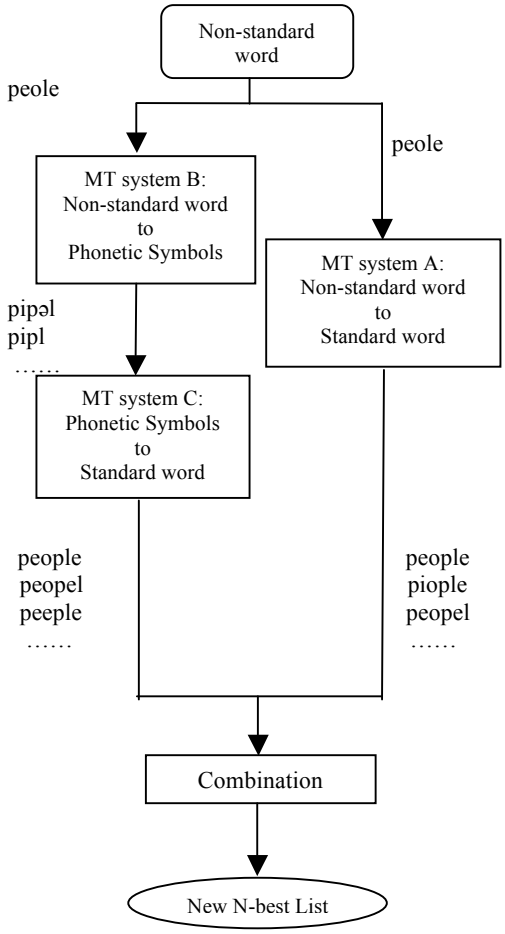

Figure 1: *Machine translation approach for normalization.*

3.2. Two-stage MT model

The above single step translation module only uses surface character information and sometimes does not properly model how non-standard words are formed. Therefore we propose to use a two-stage approach to utilize phonetic information. We notice that although there is a huge variation of possible non-standard word forms for a given standard word, a user expects that the reader can understand the non-standard form and is able to pronounce it. The simple examples are '2morrow' (tomorrow), '4ever' (forever). If there is enough training data, these can be captured by the character-based translation module, as described earlier. However, there are more difficult cases. For example, for the non-standard word 'snds', it is not easy for the single translation model to convert it to 'sounds' if not enough samples like 'sn' and 'soun' exist in the training set. For this case, maybe we can first translate 'snds' to possible phonetic sequences and then convert them back to in-vocabulary words. Therefore, we consider using pronunciation information as a bridge between the non-standard words and standard words.

This two-stage method is shown in Figure 1 (on the left). System B in this method translates a non-standard (character sequence) to its possible pronunciation (phonetic sequence).

System C translates phonetic sequence back to standard-words. Similar to the single-step method, these two modules are both accomplished using machine translation systems, where characters or phones are treated like "words". Note that in system C, the input phonetic sequence (output of system B) sometimes exists in the dictionary. For this case, we include the corresponding in-vocabulary words with high confidence score, in addition to the hypotheses from the system C. Again, similar to the single-step approach, we use a dictionary to filter the hypotheses and keep N-best candidates.

3.3. Combined System

The above two systems have different strengths and use complementary information sources, therefore we expect that combining their output will yield improved performance. As mentioned before, each of these two methods generates N-best word candidates. We use heuristic rules to combine the two lists to generate the final ranked candidates. In this process, we use the translation scores (described later) from the single-step translation module. We observe that often when the one-step translation method is confident, its result is correct; and when its confidence score is very low, the candidate is rarely a good one or it often appears on the candidate list from the two-stage approach. Therefore for the candidate list generated by the single-step translation module, we first remove words whose confidence scores are lower than a predefined threshold. Then we merge this new list with that one from the two-stage system: choose the first candidate from the single-step translation results, then the first candidate in the two-stage translation results; following is the second one in the single-step translation output, and so on, until one list is empty and the rest of the other is added. We remove duplicate candidates in the final list. We will show in the experiments that this combination method results in better system performance.

4. Experiments

4.1. Experimental Setup

For testing, we use the data set originally used by [8]. There are 303 isolated non-standard words from SMS messages, annotated with their corresponding standard words. This data has been used by various previous work for normalization. Using this common data allows us to compare our results with others.

Different parallel corpora are needed for training the translation models used in our proposed statistical machine translation system. For the single step translation system (A in Figure 1), a parallel corpus of non-standard words and standard words is needed. For this, we use the annotated data from [3], which contains 6,532 pairs of non-standard words and standard words, annotated from 4,660 Twitter messages. We call this training data set A. Note that Twitter messages share a lot of similarities with SMS since both have length limits and informal writing style.

For translation system C (translating phonetic symbol to words) in the two-stage approach, we need parallel data of words and its phonetic sequence (pronunciation). A dictionary containing words and their pronunciations can serve this purpose. Since we found that the phonetic symbols used in the CMU dictionary are not consistent with the IPA symbols [18], we decided to use an offline dictionary (http://ciba.iciba.com/) to collect commonly used English words and their corresponding IPAs. This English dictionary has 75,262 entries and is called data set C.

For system B (translating non-standard words to phonetic sequence) in the two-stage approach, we need parallel training data containing non-standard words and their pronunciations (often this is the pronunciation for the corresponding normalized words). Since we already have the above two data sets, we can easily generate this training set from data set A by replacing the standard words with their IPAs obtained from set C. In this process, we did not include all the words in set A. First some standard words in A are not included in C, such as derived words 'gets', 'chatting', 'worked', hence we do not have pronunciations for them (we plan to create pronunciations automatically in the future work). Second, some standard words are phrases, for example, 'laugh of loud', 'by the way'. After removing these words, we obtain a training set B with 4,453 pairs, among which 2,215 are unique.

We use a 5-gram character language model trained from the Edinburgh Twitter corpus for both the single-step translation model and the second stage of the two-stage method. A phone-based language model is built from the IPAs in data set C, and used for the first stage translation model in the two-stage method. SRILM toolkit [19] is used to build these language models.

We use the latest version of Moses [20] for all of our experiments. Giza++ [21] is used for automatic word alignment for the three machine translation systems. We use 8 iterations in IBM model 1, 6 iterations in HMM model, and 6 iterations in IBM model 3 in the alignment. The final score from the translation model is a combination of the four sub-models: phrase translation, language model, reordering model, and the word length penalty weight. This score corresponds to probability $P(A|S)$ in Eq(2).

To generate the final ranking for the N-best candidates in each system, we combine the translation score with a word-based unigram language model built from various corpora. We only use word unigram scores here since the decoding task is for isolated words, not full messages/sentences. We first generate a large number of candidates (250) from the translation system, and then remove those OOV words and duplicates, and keep the distinct hypotheses for each system. On average there are 30 candidates for each test word. Note that the single-step translation failed to generate candidates for 5 test cases. The outputs of the two different translation systems are combined to obtain the final ranked list as described in Sec 3.3..

4.2. Experiment Result and Discussion

Table 1 shows the N-best normalization results on the 303 pairs test set using different machine translation methods: single-step character-based translation from [3] and our implementation, our proposed two-stage method, and the combined system. Our single-step translation result is better than that in [3] mainly because of different configurations in Moses. For our proposed two-stage translation approach, we notice that its top one precision is not as good as the single-step method, but as the number of candidates increases, its coverage increases greatly. The top-20 result is better than the single-step one. This can be partly explained by the phonetic information used in this method. As expected, combining the results from the two translation systems yields the best performance. It increases the top-1 precision as well as the coverage of the top-N candidates

because the candidates generated by the two systems are complementary. Looking at the results on the test set, we find that the single-step translation method failed to normalize non-standard words like 'btiful', 'lrg', 'rit', 'snd', 'rote', but the two-stage model has the correct standard words on the N-best list, and for some even has them as the first candidate ('beautiful', 'large', and 'right').

SMS Dataset (303 pairs)	Accuracy (%)			
	Top1	Top3	Top10	Top20
Pennell & Liu [3]	60.39	74.58	75.57	75.57
Single-step MT	67.01	74.58	77.59	78.59
Two-stage MT	51.83	61.54	76.25	80.27
Combination	**71.96**	**75.33**	**81.75**	**83.11**

Table 1: N-best normalization results using machine translation methods.

For a better comparison with prior work on the same data set, we list our result along with other previous ones in Table 2 for the top-1 accuracy. We also include a baseline result using the Jazzy Spell checker. We can see that all the methods in previous work significantly outperform the Jazzy Spell checker baseline. Our result is significantly better than other published results.

SMS Data (303 pairs)	Top1 Accuracy (%)
Jazzy spell checker	43.89
Liu et al. 2011	62.05
Cook et al. 2009	59.4
Choudhury et al. 2007	59.9
Combination	**71.96**

Table 2: Top-1 results of our method in comparison to other published results.

5. Conclusion and Future work

In this paper we proposed to use a two-stage character/phone translation method for text normalization, and combine it with a previously used single-step character-based translation approach. Our experiments on a commonly used test set demonstrate the superior performance of our method. In our future work, we will refine the strategy used to combine the two systems. We also plan to combine the output from the Jazzy spell checker with our translation results. Furthermore, we will perform sentence level normalization where we will incorporate word n-gram language models to rerank the candidates. We believe that the better N-best coverage from our method will be beneficial for sentence-level decoding. Last we will evaluate the impact of normalization on TTS or other language processing tasks.

6. Acknowledgment

This work is supported by DARPA under Contract No. HR0011-12-C-0016. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

7. References

[1] http://www.ctia.org/advocacy/research/index.cfm/aid/10323
[2] http://news.cnet.com/8301-1035_3-10048257-94.html
[3] D. Pennell and Y, Liu, "A character-level machine translation approach for normalization of SMS abbreviations", IJCNLP, Chiang Mai, Thailand,2011
[4] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words", *Computer Speech and Language*, 15(3), 287-333, 2001.
[5] F. J. Damerau, "A technique for computer detection and correction of spelling errors", Commun. ACM 7:171-176, 1964
[6] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", Soviet Physics Doklady 10:707, 1966
[7] K. Toutanova and R. C. Moore, "Pronunciation modeling for improved spelling correction", ACL, Philadelphia, Pennsylvania, pp.144-151, 2002.
[8] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar, and A. Basu, "Investigation and modeling of the structure of texting language", International Journal of Document Analysis and Recognition 10, 157-174, 2007.
[9] P. Cook and S. Stevenson, "An unsupervised model for text message normalization", NAACL HLT Workshop on Computational Approaches to Linguistic Creativity, Boulder, CO, pp. 71-78, 2009.
[10] D. Pennell and Y. Liu, "Normalization of text messages for text-to-speech", ICASSP, Dallas, Texas, USA, pp.4842-4845, 2010.
[11] F. Liu, F. Weng, B. Wang, and Y. Liu, "Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision", Proc. of ACL-HLT, 2011
[12] K.F. Wong and Y. Xia, "Normalization of Chinese chat language", Language Resources and Evaluation, 219-242,2008
[13] Z. Xue, D. Yin and B.D. Davison, "Normalizing Microtext", Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011
[14] A. Aw, M. Zhang, J. Xian, and J. Su, "A phrase-based statistical model for SMS text normalization", COLING/ACL, Sydney, Australia. pp.33-40, 2006
[15] C. Kobus, F. Yvon, and G. Damnati, "Normalizing SMS: Are two metaphors better than one?", 22nd International Conference on Computational Linguistics, Manchester, UK. pp. 441-448, 2008
[16] D. Contractor, T.A. Faruquie, and L.V. Subramaniam, "Unsupervised cleansing of noisy text", Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, Stroudsburg, PA, USA. pp. 189-196, 2010.
[17] B. Han, T. Baldwin, "Lexical normalization of short text messages: Make sens a# twitter", Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,2011
[18] http://en.wikipedia.org/wiki/International_Phonetic_Alphabet
[19] A. Stolcke, "SRILM-an extensible language modeling toolkit", Proceedings of the international conference on spoken language processing, pp. 901-904, 2002
[20] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E, Herbst, "Moses: open source toolkit for statistical machine translation", Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Stroudsburg, PA, USA, pp. 177-180,2007
[21] F.J. Och and N. Hermann, "A systematic comparison of various statistical alignment models", Computational Linguistics, 29(1):19–51,2003