Input Normalization for an English-to-Chinese SMS Translation System

Aw AiTi, Zhang Min, Yeo PohKhim, Fan ZhenZhen. Su Jian
Institute of Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
{aaiti,mzhang,pohkhim,zzfan,sujian}@i2r.a-star.edu.sg

Abstract

This paper describes an approach to pre-process SMS text for Machine Translation. As SMS text behaves differently from normal written text and to reduce the tremendous effort required to customize or adapt the language model of the traditional translation system to handle SMS text style, normalization is performed to moderate the irregularities in English SMS text using a noisy channel model. A mapping model is used to model the three major problems in SMS text. They are (1) substitution of word using non-standard acronym, (2) insertion of flavour word, and (3) omission of auxiliary verb and subject pronoun. Experiment results show that with normalization before translation, the rejection rate of our English-to-Chinese SMS translation for broadcasting purpose is reduced by 15.5%. We believe that the performance of normalization can be further improved with deeper linguistic processing.

1 Introduction

Short Message Service (SMS) is an increasingly popular form of communication. It is an efficient and cost-effective platform that facilitates business and social communication. Many SMS applications are developed to provide services that value-add the business processes or promote social interaction in the communities. Taking advantage of SMS and digital television technology, we develop an application to support multilingual instant messaging from mobile phone to TV screen, making watching television an interactive process of information exchange. A SMS translator is used to translate messages from one language to another. Television viewers have the choice of displaying the messages in their preferred languages.

SMS translator faces many challenges. Firstly, SMS messages render spoken language. The language used is typically colloquial and contain many phenomena such as interjections, repetitions, ellipsis. Linguistically and stylistically they differ from written language as sentences are short and poorly structured and usually ungrammatical. These difficulties require a different approach from that taken for written documents.

Secondly, due to the restricted message length and the tediousness of text input, the message is intentionally shortened with self created and not standardized abbreviations. Words are combined either deliberately or unintentionally without linguistic considerations, posing another segmentation problem. With the inclusion of colloquial English, the messages cannot be modeled directly using a formal language model.

Thirdly, the missing of punctuation and upper/lower case distinction withhold us from the information for identifying proper names, sentence beginnings and endings. All these challenges demand an appropriately designed grammar to accommodate the input variations. Table 1 shows some examples of SMS messages.

Table 1: Examples of SMS Message

Wat happen ystrd?
U go where?
Got so many car
u noe wat happen lah.
Okae…So izit ok if i concentrate more on tchrs? Or which religion i do? Sure…No prob so u email me d survey or u prnt it out n pass2me? R we still meetg up
Ystrd..wuznt exactly a test tho…No not tt one is the 1we did in e hols…Haha…Laoshi is always loud la…Oh n we need a day2do e yunnan thing k…By nxt fri

In order to simplify the core translation task and enable us to model these problematic factors separately, we incorporate a pre-processing module to moderate the SMS text. This also enables us to model the distinctive characteristics of messages within each user group and yet have a common core MT system.

1.1 Noisy Text Translation

The presence of noise is a research issue in MT domain as it affects the robustness and performance of the final system. Studies have been conducted by Grangier (2003 & 2004) on the effect of noise in IR and in text clustering. The data is extracted from media through automatic processes such as ASR and OCR which contains recognition errors representing noise in the text. The project measured the performance degradation due to noise and found moderate degradation for text clustering. It has also shown that IR can achieve acceptable performance in the presence of high amount of noise, but the presence of noise degrades significantly the automatic summarization performances.

It is well understood that preparing input text for MT system can improve the output quality and user experiences significantly. NieBen and Ney (2001) introduced transformations before and after translation to harmonize word order for improving the performance of statistical machine translation. To have good SMS translation experiences, we preprocess the input to deal with the issues discussed in previous section by removing noise and normalizing the different types of non-standard text behavior to more linguistically well-form sentences. This paper gives an overview of SMS translation system in section 2. Section 3 gives an overview of our SMS normalizer followed by section 4 on some experimental results. Section 5 concludes the paper.

2 SMS Translation System

The SMS Translation System comprises two components, a normalization module to moderate the input text and a translation module to handle the translation. Figure 1 shows the overall architecture of the SMS translation application.

2.1 Normalization Module

Input text is first pre-processed to remove extraneous text such as "…." and "okkkkk". The text is also converted into lowercase [1] and segmented into shorter sentences. Standard and well-defined SMS lingos are also replaced by their equivalences in this stage.

Normalization takes place after pre-processing to look into lexical aspects of the sentence with the attempt to detect and correct the irregularities. A post-processing stage then puts back the proper case information in the text and joins back the sentences to form the message.

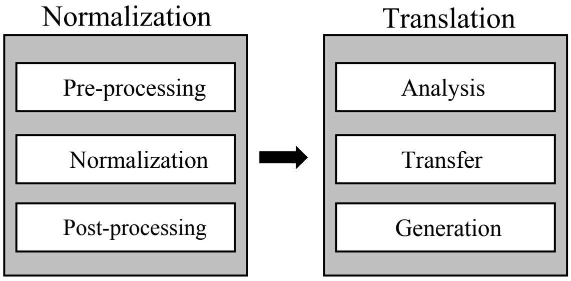


Figure 1: SMS Translation System

2.2 Translation Module

Our translation engine is a flexible rule-based system that utilizes a linguistics database in its translation process. This is a traditional transfer system involving analysis, transfer and generation, with its own set of rules and dictionaries during each step of transformation. With normalization taking place before translation, SMS messages can be translated as normal written text without changes to the language model of the MT system.

3 SMS Normalization Model

To fully convert SMS text to normal written text is not straightforward as the operation has to go beyond the detecting and correction of individual word to look into both the lexical and structural aspects of the sentence. In our system, we use a noisy channel model to perform normalization in three areas. They are (1) substitution of non-standard acronym, (2) deletion of flavour word, (3) insertion of auxiliary verb and subject pronoun.

3.1 Noisy Channel Model

Assuming the English sentence, *e*, is "corrupted" by a noisy channel to produce SMS message, *s*; the English sentence, *e*, could be recovered through a posteriori distribution for a channel target text given the source text *P(s|e)*, and a prior distribution for the channel source text *P(e)* (Shannon 1948).

$$\hat{e} = \arg\max_{e} P(s \mid e) P(e)$$

Assuming the translation model *P(s|e)* that each SMS word is translated to exactly one English word, we need to only consider two types of

[1] As uppercase/lowercase is no longer a reliable indicator of proper noun.

probabilities: the alignment probabilities denoted by $P(m|a_m)$ and the lexicon probabilities denoted by $P(s_m \mid e_{a_m})$ (Brown et al. 1993). The string translation probability can be re-written as follows.

$$P(s \mid e) = \sum_a \prod_m [P(m \mid a_m) P(s_m \mid e_{a_m})]$$

To use this model in normalizing SMS sentence, we need to extend this model to allow one word to many words alignment and vice versa. It is because lexical correspondences in our domain need to be established not only at the word level, but also at the phrase level, such as SMS lingo "*lemme*" must be corresponded with English words "*let me*" to have the most lexical affinity. We thus use a word-group channel model to support many-to-one mapping, enabling a SMS lexicon to be normalized to a sequence of adjacent English words (here referred to as word group).

3.2 Word-Group Normalization

To support many-to-one mapping, we decompose the English sentence into a sequence of word groups.

$$e_1^N = \hat{e}_1^K, \qquad \hat{e}_k = e_{n_{k-1}+1}, \ldots, e_{n_k}$$
$$s_1^M = \hat{s}_1^K, \qquad \hat{s}_k = s_m$$

We obtain the following mapping model.

$$\begin{aligned} & P(s_1^M \mid e_1^N) \\ & \approx P(\hat{s}_1^K \mid \hat{e}_1^K) \\ & = \sum_{\hat{a}} \prod_{k=1}^{K} P(k \mid \hat{a}_k) P(\hat{s}_k \mid \hat{e}_{\hat{a}_k}) \end{aligned}$$

If we consider word groups as new vocabularies in the dictionary with the inclusion of *"null"* word, we can model the three transformations directly within the word group using the mapping probability $P(\hat{s}_k \mid \hat{e}_{\hat{a}_k})$.

Insertion of article, subject pronoun and verb	$\lvert \hat{s}_k \rvert < \lvert \hat{e}_{\hat{a}_k} \rvert$
Deletion of flavour word:	$\hat{e}_{\hat{a}_k} = \text{null}$
Substitution of non-standard acronym:	$\lvert \hat{s}_k \rvert = \lvert \hat{e}_{\hat{a}_k} \rvert$

Assuming monotone alignment, our task is thus focused on the word group distributions, mainly the segmentation of English sentence such that it maximizes the following equation.

$$P(s_1^M \mid e_1^N) \approx \sum_k P(\hat{s}_k \mid \hat{e}_k)$$

The probability of each mapping $P(\hat{s}_k \mid \hat{e}_k)$ is estimated via relative frequencies as follows:

$$P(\hat{s}_k \mid \hat{e}_k) = \frac{N(\hat{s}_k, \hat{e}_k)}{\sum_{\hat{s}'} N(\hat{s}', \hat{e}_k)}$$

Here, $N(\hat{s}_k, \hat{e}_k)$ denotes the count of the event that $\hat{s}_k$ has been normalized to $\hat{e}_k$.

3.3 Initial Alignment and Training

The foremost task of computing $P(\hat{s}_k \mid \hat{e}_k)$ is to identify word groups $\hat{s}_k$ and $\hat{e}_k$ that maximize $P(s_1^M \mid e_1^N)$. As there may not exist a clear linguistic relationship among the constituents in the SMS word groups, word groups are discovered through an initial force alignment strategy and iterated through the EM algorithm (Dempster, 1977). Theoretically a word group can be formed with any combination of its adjacent words.

```
The Expectation-Maximization Algorithm
(1) Bootstrap using initial Alignment
(2) Expectation: Update mapping model
(3) Maximization: Apply mapping model to get
    new alignment
(4) Repeat (2) to (3) until mapping converges
```

To accelerate the convergence of EM training and reduce noisy aligned pairs $(\hat{s}_k, \hat{e}_k)$, we bootstrap mapping based on orthographic similarity and with the help of a SMS lingo dictionary. In searching, correspondence boundary candidates which satisfy the above matching predicates are first established. Simple heuristics are employed to match lexicons within the pairs of boundary candidates. Consecutive lexicons within the boundary candidates are being dynamically combined as candidate word group if their lengths do not agree. To further reduce search time, deletion is assumed if correspondence boundary candidates fail to establish within the vicinity of 5 lexicons. Table 2 shows the word group extraction strategy.

```
(1) Form boundary candidates if
    a)string_compare($e_n$,$s_m$)==0
            ;exact string match
    b)short_form($e_n$)== $s_m$
            ; $s_m$ is a standard lingo of $e_n$
    c)string_similarity($e_n$, $s_m$) > $\alpha$
            ; string similarity is defined using %
    of common letters

(2) For each segment formed by boundary
candidates

    Case 1
    if token length of SMS Segment > token
    length of English Segment

    - map SMS boundary candidates to English
        boundary candidates
    - from left to right, map one SMS token to one
        English token
    - map remaining SMS tokens to null

    Case 2
    if token length of SMS Segment = token
    length of English Segment

    - map one SMS token to one English token

    Case 3
    if token length of SMS Segment < token
    length of English Segment

    -   from left to right, form word group by
        combining leftmost tokens until the two
        segments have same token length
    -   from left to right, map one SMS token to
        one English token
```

Table 2: Word Group Extraction Strategy

There is no doubt that pure statistics cannot perform very reliably. Some word groups found by the algorithm are awkward to be accepted as word group unit. Refinement of mapping pairs was carried out manually to remove superfluous entries. Table 3 shows some examples of mapping pairs.

$\hat{s}$	$\hat{e}$	$\log P(\hat{s} \mid \hat{e})$
2	2 two to too Null	0 -0.0791812 -0.579466 -0.897016 -2.97058
Cu	see you	0
Dat	that Date	-0.726999 -0.845098
Tmr	tomorrow	-0.341514
2morro	tomorrow	-2.28103
Tmrw	tomorrow	-1.98

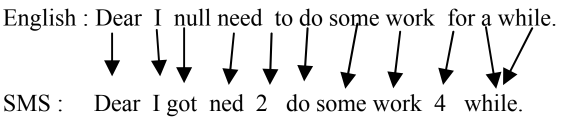
Table 3: Examples of Mapping pairs

3.4 Decoding

The set of mapping pairs derived from above alignment process forms the mapping table which is then used in our normalization decoding. As the decoder is bounded by this table, it is important that the training database covers as much as possible the potential mapping patterns.

The Viterbi algorithm is used to produce the best sentence by maximizing the overall probability.

$$P(e) \approx \prod_{k=1}^{K} [P(\hat{s}_k \mid \hat{e}_k) \prod_{i=n_{k-1}+1}^{n_k} P(e_i \mid e_{i-1})]$$


English : Dear I null need to do some work for a while.
SMS : Dear I got ned 2 do some work 4 while.

Mapping	Transformation
null → got	deletion
need → ned to → 2 for → 4	substitution
a while → while	insertion

Figure 2. An example of decoding

Due to the unavailability of large SMS corpus, our n-gram language model is trained on extracts from English Gigaword provided by LDC using SRILM language modelling toolkit. Backoff smoothing (Jelink, 1991) is used to adjust and assign a non-zero probability to the unseen words to address data sparseness. Figure 2 shows a decoding example.

4 The Experiment

4.1 Normalization Results

A training set of 5162 parallel SMS messages extracted from our corpus collection is prepared manually by the project group. Both open and close tests are performed using five-fold cross validation. The results are then compared with the manually normalized data (reference) using the following indicators.

		Reference r_i	
		Changed	Unchanged
Normalized n_j	Changed	[$r_i = n_j$] correct *COR* *INC* Incorrect [$r_i \neq n_j$]	spurious *SPU*
	Unchanged	missing *MIS*	non-committal *NON*

The overall score is measured using precision and recall, which is defined as follows and calculated based on each correction made by the system.

Normalization Accuracy(PRE)
= COR / COR + INC + SPU

Normalization Relevancy (REC)
= COR / COR + INC + MIS

In the experiment of SMS normalization, the proposed method achieves on average of 71.9% normalization accuracy and 79.4% normalization relevancy for open test and 79.7% normalization accuracy and 79.9% normalization relevancy for close test using a bigram language model. The performance, however, weakens with a trigram model with 64.0% normalization accuracy and 74.6% normalization relevancy for open test and 76.0% normalization accuracy and 77.3% normalization relevancy for close test.

The behaviour is likely due to the fact that the language model is trained using newspaper texts instead of SMS texts, and thus higher-order language model cannot model the context as effectively as compared to a lower-order model. An investigation to the results also reveals that the model is most effective in handling common abbreviation substitution and performs poorest for insertion. It is observed that mapping such as "*you*" to "*are you*" or "*you are*" could not be effectively modelled in the test set. Insertion is generally a more complex problem than substitution and demands a higher-order n-gram model and a larger amount of training data. The use of formal text as training data can also contribute to the poor performance in insertion.

4.2 Effect on Translation

No published results have been reported so far on SMS text normalization and its effect on SMS Translation. We conducted experiment to find out its effect on 200 messages randomly selected from the text corpus. The translation results are then evaluated manually to determine the effort required to post-edit the output for dissemination purpose. It shows that we are able to increase the acceptance rate of the translation output by 15.5% by minor post-editing through input normalization.

	Acceptance Rate		Rejection Rate[2]
	Without Post-edit[3]	*With Post-edit*[4]	
Translation	45.5%	27.5%	27.0%
Normalization + Translation	58.5%	30.0%	11.5%

5 Conclusion

In this paper, we propose a supervised learning approach to insert, delete and substitute words on SMS text. The proposed approach attempts to normalize the SMS text style so that some form of adherence to the norm of written language can be achieved before translation takes place. The approach automatically learns the parameters of the model from parallel SMS texts and does not rely on any human-maintained resources.

Though the development corpus is relatively small due to the unavailability and difficulties of getting such corpora, the results obtained nevertheless provide us with a good indication on the feasibility of using this approach in performing the task. We plan to extend the experiments to incorporate higher level syntactic processing using a bigger data set and explore new approaches that

[2] The translation cannot be used directly and requires major rephrasing or rewriting of the whole translation.

[3] The translation is good and can be broadcasted to the public without modifications.

[4] The translation is readable and is good after minor correction.

lead to more accurate alignment and normalization.

6 Acknowledgements

This work was partially funded by the Media Development Authority of Singapore under the Digital Technology Development Scheme.

References

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, J. D. Lafferty, R. L. Mercer, H. Printz, L. Ures. *The Candide System for Machine Translation (1994) In Proc.* , ARPA Workshop on Human Language Technology, pages 157–162.

E. Brill & R. C. Moore. *An Improved Error Model for Noisy Channel Spelling Correction*. ACL 2000.

A. Clark. *Pre-processing very noisy text.* Workshop on Shallow Processing of Large Corpora, SProLaC 2003

Dempster, A.P.,N.M. Laird and D.B.Rubin, 1977. *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Stat. Soc., Ser.B. Vol. 39, pp138. F. Mosteller and D. Wallace. 1964. Inference and Disputed Authorship: The Federalist. Addison-Wesley, Reading, Massachusetts.

D. Grangier, A. Vinciarelli, 2004. *Noisy Text Clustering*. IDIAP-RR 04-31.

D. Grangier, A. Vinciarelli, H. Bourlard, 2004. *Information Retrieval on Noisy Text.* IDIAP-COM 03-08.

H. Z. Li, M. Zhang, J. Su. *A Joint Source-Channel Model for Machine Transliteration*, ACL 2004

S. Nießen, H. Ney. *Morpho-Syntactic Analysis for Reordering in Statistical Machine Translation*. eamt.org/summitVIII

F. J. Och, C. Tillman, and H. Ney, *Improved Alignment Models for Statistical Machine Translation*, Proc. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999

K. Papineni, S. Roukos, T. Ward, Wei-Jing Zhu, 2001. *Bleu: a Method for Automatic Evaluation of Machine Translation*.

E. K. Ringger & J. F. Allen. *A Fertility Channel Model for Post-Correction of Con-tinuous Speech Recognition.* ICSLP 1996.

J. Tomas, F. Casacuberta. *Monotone Statistical Translation Using Word Group*. MT Summit VIII, 2001

K. Toutanova & R. C. Moore. *Pronunciation Modeling for Improved Spelling Correction*. ACL 2002