A Phonetic-Based Approach to Chinese Chat Text Normalization

Yunqing Xia, Kam-Fai Wong
Department of S.E.E.M.
The Chinese University of Hong Kong
Shatin, Hong Kong
{yqxia, kfwong}@se.cuhk.edu.hk

Wenjie Li
Department of Computing
The Hong Kong Polytechnic University
Kowloon, Hong Kong
cswjli@comp.polyu.edu.hk

Abstract

Chatting is a popular communication media on the Internet via ICQ, chat rooms, etc. Chat language is different from natural language due to its anomalous and dynamic natures, which renders conventional NLP tools inapplicable. The dynamic problem is enormously troublesome because it makes static chat language corpus outdated quickly in representing contemporary chat language. To address the dynamic problem, we propose the phonetic mapping models to present mappings between chat terms and standard words via phonetic transcription, i.e. Chinese Pinyin in our case. Different from character mappings, the phonetic mappings can be constructed from available standard Chinese corpus. To perform the task of dynamic chat language term normalization, we extend the source channel model by incorporating the phonetic mapping models. Experimental results show that this method is effective and stable in normalizing dynamic chat language terms.

1 Introduction

Internet facilitates online chatting by providing ICQ, chat rooms, BBS, email, blogs, etc. Chat language becomes ubiquitous due to the rapid proliferation of Internet applications. Chat language text appears frequently in chat logs of online education (Heard-White, 2004), customer relationship management (Gianforte, 2003), etc. On the other hand, wed-based chat rooms and BBS systems are often abused by solicitors of terrorism, pornography and crime (McCullagh, 2004). Thus there is a social urgency to understand online chat language text.

Chat language is anomalous and dynamic. Many words in chat text are anomalous to natural language. Chat text comprises of ill-edited terms and anomalous writing styles. We refer chat terms to the anomalous words in chat text. The dynamic nature reflects that chat language changes more frequently than natural languages. For example, many popular chat terms used in last year have been discarded and replaced by new ones in this year. Details on these two features are provided in Section 2.

The anomalous nature of Chinese chat language is investigated in (Xia et al., 2005). Pattern matching and SVM are proposed to recognize the ambiguous chat terms. Experiments show that F-1 measure of recognition reaches 87.1% with the biggest training set. However, it is also disclosed that quality of both methods drops significantly when training set is older. The dynamic nature is investigated in (Xia et al., 2006a), in which an error-driven approach is proposed to detect chat terms in dynamic Chinese chat terms by combining standard Chinese corpora and NIL corpus (Xia et al., 2006b). Language texts in standard Chinese corpora are used as negative samples and chat text pieces in the NIL corpus as positive ones. The approach calculates confidence and entropy values for the input text. Then threshold values estimated from the training data are applied to identify chat terms. Performance equivalent to the methods in existence is achieved consistently. However, the issue of normalization is addressed in their work. Dictionary based chat term normalization is not a good solution because the dictionary cannot cover new chat terms appearing in the dynamic chat language.

In the early stage of this work, a method based on source channel model is implemented for chat term normalization. The problem we encounter is addressed as follows. To deal with the anomalous nature, a chat language corpus is constructed with chat text collected from the Internet. How-

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 993–1000,
Sydney, July 2006.

ever, the dynamic nature renders the static corpus outdated quickly in representing contemporary chat language. The dilemma is that timely chat language corpus is nearly impossible to obtain. The sparse data problem and dynamic problem become crucial in chat term normalization. We believe that some information beyond character should be discovered to help addressing these two problems.

Observation on chat language text reveals that most Chinese chat terms are created via phonetic transcription, i.e. Chinese Pinyin in our case. A more exciting finding is that the phonetic mappings between standard Chinese words and chat terms remain stable in dynamic chat language. We are thus enlightened to make use of the phonetic mapping models, in stead of character mapping models, to design a normalization algorithm to translate chat terms to their standard counterparts. Different from the character mapping models constructed from chat language corpus, the phonetic mapping models are learned from a standard language corpus because they attempt to model mappings probabilities between any two Chinese characters in terms of phonetic transcription. Now the sparse data problem can thus be appropriately addressed. To normalize the dynamic chat language text, we extend the source channel model by incorporating phonetic mapping models. We believe that the dynamic problem can be resolved effectively and robustly because the phonetic mapping models are stable.

The remaining sections of this paper are organized as follows. In Section 2, features of chat language are analyzed with evidences. In Section 3, we present methodology and problems of the source channel model approach to chat term normalization. In Section 4, we present definition, justification, formalization and parameter estimation for the phonetic mapping model. In Section 5, we present the extended source channel model that incorporates the phonetic mapping models. Experiments and results are presented in Section 6 as well as discussions and error analysis. We conclude this paper in Section 7.

2 Feature Analysis and Evidences

Observation on NIL corpus discloses the anomalous and dynamic features of chat language.

2.1 Anomalous

Chat language is explicitly anomalous in two aspects. Firstly, some chat terms are anomalous entries to standard dictionaries. For example, "介里(here, *jie4 li3*)" is not a standard word in any contemporary Chinese dictionary while it is often used to replace "这里(here, *zhe4 li3*)" in chat language. Secondly, some chat terms can be found in standard dictionaries while their meanings in chat language are anomalous to the dictionaries. For example, "偶(even, *ou3*)" is often used to replace "我(me, *wo2*)" in chat text. But the entry that "偶" occupies in standard dictionary is used to describe even numbers. The latter case is constantly found in chat text, which makes chat text understanding fairly ambiguous because it is difficult to find out whether these terms are used as standard words or chat terms.

2.2 Dynamic

Chat text is deemed dynamic due to the fact that a large proportion of chat terms used in last year may become obsolete in this year. On the other hand, ample new chat terms are born. This feature is not as explicit as the anomalous nature. But it is as crucial. Observation on chat text in NIL corpus reveals that chat term set changes along with time very quickly.

An empirical study is conducted on five chat text collections extracted from YESKY BBS system (bbs.yesky.com) within different time periods, i.e. Jan. 2004, July 2004, Jan. 2005, July 2005 and Jan. 2006. Chat terms in each collection are picked out by hand together with their frequencies so that five chat term sets are obtained. The top 500 chat terms with biggest frequencies in each set are selected to calculate reoccurring rates of the earlier chat term sets on the later ones.

Set	Jul-04	Jan-05	Jul-05	Jan-06	Avg.
Jan-04	0.882	0.823	0.769	**0.706**	0.795
Jul-04	-	0.885	0.805	0.749	0.813
Jan-05	-	-	0.891	0.816	0.854
Jul-05	-	-	-	0.875	0.875

Table 1. Chat term re-occurring rates. The rows represent the earlier chat term sets and the columns the later ones.

The surprising finding in Table 1 is that 29.4% of chat terms are replaced with new ones within two years and about 18.5% within one year. The changing speed is much faster than that in standard language. This thus proves that chat text is dynamic indeed. The dynamic nature renders the static corpus outdated quickly. It poses a challenging issue on chat language processing.

3 Source Channel Model and Problems

The source channel model is implemented as baseline method in this work for chat term normalization. We brief its methodology and problems as follows.

3.1 The Model

The source channel model (SCM) is a successful statistical approach in speech recognition and machine translation (Brown, 1990). SCM is deemed applicable to chat term normalization due to similar task nature. In our case, SCM aims to find the character string $C = \{c_i\}_{i=1,2,...,n}$ that the given input chat text $T = \{t_i\}_{j=1,2,...,n}$ is most probably translated to, i.e. $t_i \rightarrow c_i$, as follows.

$$\hat{C} = \arg\max_C p(C \mid T) = \arg\max_C \frac{p(T \mid C)p(C)}{p(T)} \quad (1)$$

Since $p(T)$ is a constant for C, so $\hat{C}$ should also maximize $p(T \mid C)p(C)$. Now $p(C \mid T)$ is decomposed into two components, i.e. chat term translation observation model $p(T \mid C)$ and language model $p(C)$. The two models can be both estimated with maximum likelihood method using the trigram model in NIL corpus.

3.2 Problems

Two problems are notable in applying SCM in chat term normalization. First, data sparseness problem is serious because timely chat language corpus is expensive thus small due to dynamic nature of chat language. NIL corpus contains only 12,112 pieces of chat text created in eight months, which is far from sufficient to train the chat term translation model. Second, training effectiveness is poor due to the dynamic nature. Trained on static chat text pieces, the SCM approach would perform poorly in processing chat text in the future. Robustness on dynamic chat text thus becomes a challenging issue in our research.

Updating the corpus with recent chat text constantly is obviously not a good solution to the above problems. We need to find some information beyond character to help addressing the sparse data problem and dynamic problem. Fortunately, observation on chat terms provides us convincing evidence that the underlying phonetic mappings exist between most chat terms and their standard counterparts. The phonetic mappings are found promising in resolving the two problems.

4 Phonetic Mapping Model

4.1 Definition of Phonetic Mapping

Phonetic mapping is the bridge that connects two Chinese characters via phonetic transcription, i.e. Chinese Pinyin in our case. For example, "介 $\xrightarrow{(zhe,\ jie,0.56)}$ 这" is the phonetic mapping connecting "这(this, *zhe4*)" and "介(interrupt, *jie4*)", in which "*zhe*" and "*jie*" are Chinese Pinyin for "这" and "介" respectively. 0.56 is phonetic similarity between the two Chinese characters. Technically, the phonetic mappings can be constructed between any two Chinese characters within any Chinese corpus. In chat language, any Chinese character can be used in chat terms, and phonetic mappings are applied to connect chat terms to their standard counterparts. Different from the dynamic character mappings, the phonetic mappings can be produced with standard Chinese corpus before hand. They are thus stable over time.

4.2 Justifications on Phonetic Assumption

To make use of phonetic mappings in normalization of chat language terms, an assumption must be made that chat terms are mainly formed via phonetic mappings. To justify the assumption, two questions must be answered. First, how many percent of chat terms are created via phonetic mappings? Second, why are the phonetic mapping models more stable than character mapping models in chat language?

Mapping type	Count	Percentage
Chinese word/phrase	9370	83.3%
English capital	2119	7.9%
Arabic number	1021	8.0%
Other	1034	0.8%

Table 2. Chat term distribution in terms of mapping type.

To answer the first question, we look into chat term distribution in terms of mapping type in Table 2. It is revealed that 99.2 percent of chat terms in NIL corpus fall into the first four phonetic mapping types that make use of phonetic mappings. In other words, 99.2 percent of chat terms can be represented by phonetic mappings. 0.8% chat terms come from the OTHER type, emoticons for instance. The first question is undoubtedly answered with the above statistics.

To answer the second question, an observation is conducted again on the five chat term sets described in Section 2.2. We create phonetic map-

pings manually for the 500 chat terms in each set. Then five phonetic mapping sets are obtained. They are in turn compared against the standard phonetic mapping set constructed with Chinese Gigaword. Percentage of phonetic mappings in each set covered by the standard set is presented in Table 3.

Set	Jan-04	Jul-04	Jan-05	Jul-05	Jan-06
percentage	98.7	99.3	98.9	99.3	99.1

Table 3. Percentages of phonetic mappings in each set covered by standard set.

By comparing Table 1 and Table 3, we find that phonetic mappings remain more stable than character mappings in chat language text. This finding is convincing to justify our intention to design effective and robust chat language normalization method by introducing phonetic mappings to the source channel model. Note that about 1% loss in these percentages comes from chat terms that are not formed via phonetic mappings, emoticons for example.

4.3 Formalism

The phonetic mapping model is a five-tuple, i.e.

$$< T, C, pt(T), pt(C), \Pr_{pm}(T \mid C) >,$$

which comprises of chat term character T, standard counterpart character C, phonetic transcription of T and C, i.e. $pt(T)$ and $pt(C)$, and the mapping probability $\Pr_{pm}(T \mid C)$ that T is mapped to C via the phonetic mapping $T \xrightarrow{(pt(T), pt(C), \Pr_{pm}(T|C))} C$ (hereafter briefed by $T \xrightarrow{M} C$).

As they manage mappings between any two Chinese characters, the phonetic mapping models should be constructed with a standard language corpus. This results in two advantages. One, sparse data problem can be addressed appropriately because standard language corpus is used. Two, the phonetic mapping models are as stable as standard language. In chat term normalization, when the phonetic mapping models are used to represent mappings between chat term characters and standard counterpart characters, the dynamic problem can be addressed in a robust manner.

Differently, the character mapping model used in the SCM (see Section 3.1) connects two Chinese characters directly. It is a three-tuple, i.e.

$$< T, C, \Pr_{cm}(T \mid C) >,$$

which comprises of chat term character T, standard counterpart character C and the mapping probability $\Pr_{cm}(T \mid C)$ that T is mapped to C via this character mapping. As they must be constructed from chat language training samples, the character mapping models suffer from data sparseness problem and dynamic problem.

4.4 Parameter Estimation

Two questions should be answered in parameter estimation. First, how are the phonetic mapping space constructed? Second, how are the phonetic mapping probabilities estimated?

To construct the phonetic mapping models, we first extract all Chinese characters from standard Chinese corpus and use them to form candidate character mapping models. Then we generate phonetic transcription for the Chinese characters and calculate phonetic probability for each candidate character mapping model. We exclude those character mapping models holding zero probability. Finally, the character mapping models are converted to phonetic mapping models with phonetic transcription and phonetic probability incorporated.

The phonetic probability is calculated by combining phonetic similarity and character frequencies in standard language as follows.

$$\Pr ob_{pm}(A, \overline{A}) = \frac{\left(fr_{slc}(\overline{A}) \times ps(A, \overline{A})\right)}{\sum_i \left(fr_{slc}(A_i) \times ps(A, A_i)\right)} \quad (2)$$

In Equation (2) $\{A_i\}$ is the character set in which each element A_i is similar to character A in terms of phonetic transcription. $fr_{slc}(c)$ is a function returning frequency of given character c in standard language corpus and $ps(c_1, c_2)$ phonetic similarity between character c_1 and c_2.

Phonetic similarity between two Chinese characters is calculated based on Chinese Pinyin as follows.

$$\begin{aligned} ps(A, \overline{A}) &= Sim(py(A), py(\overline{A})) \\ &= Sim(initial(py(A)), initial(py(\overline{A}))) \\ &\times Sim(final(py(A)), final(py(\overline{A}))) \end{aligned} \quad (3)$$

In Equation (3) $py(c)$ is a function that returns Chinese Pinyin of given character c, and $initial(x)$ and $final(x)$ return initial (*shengmu*) and final (*yunmu*) of given Chinese Pinyin x respectively. For example, Chinese Pinyin for the Chinese character "这" is "*zhe*", in which "*zh*" is initial and "*e*" is final. When initial or final is

empty for some Chinese characters, we only calculate similarity of the existing parts.

An algorithm for calculating similarity of initial pairs and final pairs is proposed in (Li et al., 2003) based on letter matching. Problem of this algorithm is that it always assigns zero similarity to those pairs containing no common letter. For example, initial similarity between "*ch*" and "*q*" is set to zero with this algorithm. But in fact, pronunciations of the two initials are very close to each other in Chinese speech. So non-zero similarity values should be assigned to these special pairs before hand (e.g., similarity between "*ch*" and "*q*" is set to 0.8). The similarity values are agreed by some native Chinese speakers. Thus Li et al.'s algorithm is extended to output a pre-defined similarity value before letter matching is executed in the original algorithm. For example, Pinyin similarity between "*chi*" and "*qi*" is calculated as follows.

$$Sim(chi,qi) = Sim(ch,q) \times Sim(i,i) = 0.8 \times 1 = 0.8$$

5 Extended Source Channel Model

We extend the source channel model by inserting phonetic mapping models $M = \{m_i\}_{i=1,2,...,n}$ into equation (1), in which chat term character t_i is mapped to standard character c_i via m_i, i.e. $t_i \xrightarrow{m_i} c_i$. The extended source channel model (XSCM) is mathematically addressed as follows.

$$\begin{aligned} \hat{C} &= \arg\max_{C,M} \; p(C \mid M,T) \\ &= \arg\max_{C,M} \; \frac{p(T \mid M,C)\,p(M \mid C)\,p(C)}{p(T)} \end{aligned} \quad (4)$$

Since $p(T)$ is a constant, $\hat{C}$ and $\hat{M}$ should also maximize $p(T \mid M,C)p(M \mid C)p(C)$. Now three components are involved in XSCM, i.e. chat term normalization observation model $p(T \mid M,C)$, phonetic mapping model $p(M \mid C)$ and language model $p(C)$.

Chat Term Normalization Observation Model. We assume that mappings between chat terms and their standard Chinese counterparts are independent of each other. Thus chat term normalization probability can be calculated as follows.

$$p(T \mid M,C) = \prod_i p(t_i \mid m_i, c_i) \quad (5)$$

The $p(t_i \mid m_i, c_i)$'s are estimated using maximum likelihood estimation method with Chinese character trigram model in NIL corpus.

Phonetic Mapping Model. We assume that the phonetic mapping models depend merely on the current observation. Thus the phonetic mapping probability is calculated as follows.

$$p(M \mid C) = \prod_i p(m_i \mid c_i) \quad (6)$$

in which $p(m_i \mid c_i)$'s are estimated with equation (2) and (3) using a standard Chinese corpus.

Language Model. The language model $p(C)$'s can be estimated using maximum likelihood estimation method with Chinese character trigram model on NIL corpus.

In our implementation, Katz Backoff smoothing technique (Katz, 1987) is used to handle the sparse data problem, and Viterbi algorithm is employed to find the optimal solution in XSCM.

6 Evaluation

6.1 Data Description

Training Sets

Two types of training data are used in our experiments. We use news from Xinhua News Agency in LDC Chinese Gigaword v.2 (CNGIGA) (Graf et al., 2005) as standard Chinese corpus to construct phonetic mapping models because of its excellent coverage of standard Simplified Chinese. We use NIL corpus (Xia et al., 2006b) as chat language corpus. To evaluate our methods on size-varying training data, six chat language corpora are created based on NIL corpus. We select 6056 sentences from NIL corpus randomly to make the first chat language corpus, i.e. C#1. In every next corpus, we add extra 1,211 random sentences. So 7,267 sentences are contained in C#2, 8,478 in C#3, 9,689 in C#4, 10,200 in C#5, and 12,113 in C#6.

Test Sets

Test sets are used to prove that chat language is dynamic and XSCM is effective and robust in normalizing dynamic chat language terms. Six time-varying test sets, i.e. T#1 ~ T#6, are created in our experiments. They contain chat language sentences posted from August 2005 to Jan 2006. We randomly extract 1,000 chat language sentences posted in each month. So timestamp of the six test sets are in temporal order, in which timestamp of T#1 is the earliest and that of T#6 the newest.

The normalized sentences are created by hand and used as standard normalization answers.

6.2 Evaluation Criteria

We evaluate two tasks in our experiments, i.e. recognition and normalization. In recognition, we use precision (*p*), recall (*r*) and f-1 measure (*f*) defined as follows.

$$p = \frac{x}{x+y} \quad r = \frac{x}{x+z} \quad f = \frac{2 \times p \times r}{p+r} \qquad (7)$$

where *x* denotes the number of true positives, *y* the false positives and *z* the true negatives.

For normalization, we use accuracy (*a*), which is commonly accepted by machine translation researchers as a standard evaluation criterion. Every output of the normalization methods is compared to the standard answer so that normalization accuracy on each test set is produced.

6.3 Experiment I: SCM vs. XSCM Using Size-varying Chat Language Corpora

In this experiment we investigate on quality of XSCM and SCM using same size-varying training data. We intend to prove that chat language is dynamic and phonetic mapping models used in XSCM are helpful in addressing the dynamic problem. As no standard Chinese corpus is used in this experiment, we use standard Chinese text in chat language corpora to construct phonetic mapping models in XSCM. This violates the basic assumption that the phonetic mapping models should be constructed with standard Chinese corpus. So results in this experiment should be used only for comparison purpose. It would be unfair to make any conclusion on general performance of XSCM method based on results in this experiments.

We train the two methods with each of the six chat language corpora, i.e. C#1 ~ C#6 and test them on six time-varying test sets, i.e. T#1 ~ T#6. F-1 measure values produced by SCM and XSCM in this experiment are present in Table 3.

Three tendencies should be pointed out according to Table 3. The first tendency is that f-1 measure in both methods drops on time-varying test sets (see Figure 1) using same training chat language corpora. For example, both SCM and XSCM perform best on the earliest test set T#1 and worst on newest T#4. We find that the quality drop is caused by the dynamic nature of chat language. It is thus revealed that chat language is indeed dynamic. We also find that quality of XSCM drops less than that of SCM. This proves that phonetic mapping models used in XSCM are helpful in addressing the dynamic problem. However, quality of XSCM in this experiment still drops by 0.05 on the six time-varying test sets. This is because chat language text corpus is used as standard language corpus to model the phonetic mappings. Phonetic mapping models constructed with chat language corpus are far from sufficient. We will investigate in Experiment-II to prove that stable phonetic mapping models can be constructed with real standard language corpus, i.e. CNGIGA.

Test Set		T#1	T#2	T#3	T#4	T#5	T#6
SCM	C#1	0.829	0.805	0.762	0.701	0.739	0.705
	C#2	0.831	0.807	0.767	0.711	0.745	0.715
	C#3	0.834	0.811	0.774	0.722	0.751	0.722
	C#4	0.835	0.814	0.779	0.729	0.753	0.729
	C#5	0.838	0.816	0.784	0.737	0.761	0.737
	C#6	0.839	0.819	0.789	0.743	0.765	0.743
XSCM	C#1	0.849	0.840	0.820	0.790	0.805	0.790
	C#2	0.850	0.841	0.824	0.798	0.809	0.796
	C#3	0.850	0.843	0.824	0.797	0.815	0.800
	C#4	0.851	0.844	0.829	0.805	0.819	0.805
	C#5	0.852	0.846	0.833	0.811	0.823	0.811
	C#6	0.854	0.849	0.837	0.816	0.827	0.816

Table 3. F-1 measure by SCM and XSCM on six test sets with six chat language corpora.

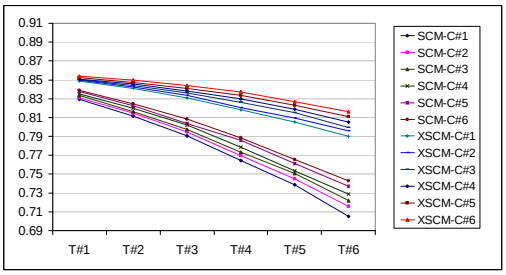

Figure 1. Tendency on f-1 measure in SCM and XSCM on six test sets with six chat language corpora.

The second tendency is f-1 measure of both methods on same test sets drops when trained with size-varying chat language corpora. For example, both SCM and XSCM perform best on the largest training chat language corpus C#6 and worst on the smallest corpus C#1. This tendency reveals that both methods favor bigger training chat language corpus. So extending the chat language corpus should be one choice to improve quality of chat language term normalization.

The last tendency is found on quality gap between SCM and XSCM. We calculate f-1 measure gaps between two methods using same training sets on same test sets (see Figure 2). Then the tendency is made clear. Quality gap between SCM and XSCM becomes bigger when test set

becomes newer. On the oldest test set T#1, the gap is smallest, while on the newest test set T#6, the gap reaches biggest value, i.e. around 0.09. This tendency reveals excellent capability of XSCM in addressing dynamic problem using the phonetic mapping models.

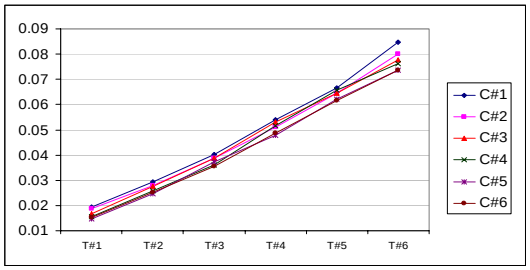

Figure 2. Tendency on f-1 measure gap in SCM and XSCM on six test sets with six chat language corpora.

6.4 Experiment II: SCM vs. XSCM Using Size-varying Chat Language Corpora and CNGIGA

In this experiment we investigate on quality of SCM and XSCM when a real standard Chinese language corpus is incorporated. We want to prove that the dynamic problem can be addressed effectively and robustly when CNGIGA is used as standard Chinese corpus.

We train the two methods on CNGIGA and each of the six chat language corpora, i.e. C#1 ~ C#6. We then test the two methods on six time-varying test sets, i.e. T#1 ~ T#6. F-1 measure values produced by SCM and XSCM in this experiment are present in Table 4.

Test Set		T#1	T#2	T#3	T#4	T#5	T#6
SCM	C#1	0.849	0.840	0.820	0.790	0.735	0.703
	C#2	0.850	0.841	0.824	0.798	0.743	0.714
	C#3	0.850	0.843	0.824	0.797	0.747	0.720
	C#4	0.851	0.844	0.829	0.805	0.748	0.727
	C#5	0.852	0.846	0.833	0.811	0.758	0.734
	C#6	0.854	0.849	0.837	0.816	0.763	0.740
XSCM	C#1	0.880	0.878	0.883	0.878	0.881	0.878
	C#2	0.883	0.883	0.888	0.882	0.884	0.880
	C#3	0.885	0.885	0.890	0.884	0.887	0.883
	C#4	0.890	0.888	0.893	0.888	0.893	0.887
	C#5	0.893	0.892	0.897	0.892	0.897	0.892
	C#6	0.898	0.896	0.900	0.897	0.901	0.896

Table 4. F-1 measure by SCM and XSCM on six test sets with six chat language corpora and CNGIGA.

Three observations are conducted on our results. First, according to Table 4, f-1 measure of SCM with same training chat language corpora drops on time-varying test sets, but XSCM produces much better f-1 measure consistently using CNGIGA and same training chat language corpora (see Figure 3). This proves that phonetic mapping models are helpful in XSCM method. The phonetic mapping models contribute in two aspects. On the one hand, they improve quality of chat term normalization on individual test sets. On the other hand, satisfactory robustness is achieved consistently.

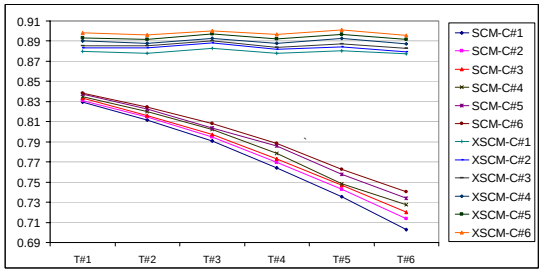

Figure 3. Tendency on f-1 measure in SCM and XSCM on six test sets with six chat language corpora and CNGIGA.

The second observation is conducted on phonetic mapping models constructed with CNGIGA. We find that 4,056,766 phonetic mapping models are constructed in this experiment, while only 1,303,227 models are constructed with NIL corpus in Experiment I. This reveals that coverage of standard Chinese corpus is crucial to phonetic mapping modeling. We then compare two character lists constructed with two corpora. The 100 characters most frequently used in NIL corpus are rather different from those extracted from CNGIGA. We can conclude that phonetic mapping models should be constructed with a sound corpus that can represent standard language.

The last observation is conducted on f-1 measure achieved by same methods on same test sets using size-varying training chat language corpora. Both methods produce best f-1 measure with biggest training chat language corpus C#6 on same test sets. This again proves that bigger training chat language corpus could be helpful to improve quality of chat language term normalization. One question might be asked whether quality of XSCM converges on size of the training chat language corpus. This question remains open due to limited chat language corpus available to us.

6.5 Error Analysis

Typical errors in our experiments belong mainly to the following two types.

Err.1 Ambiguous chat terms
Example-1: 我还是 8 米

In this example, XSCM finds no chat term while the correct normalization answer is "我还是不明 (I still don't understand)". Error illustrated in Example-1 occurs when chat terms "8(eight, *ba1*)" and "米(meter, *mi3*)" appear in a chat sentence together. In chat language, "米" in some cases is used to replace "明(understand, *ming2*)", while in other cases, it is used to represent a unit for length, i.e. meter. When number "8" appears before "米", it is difficult to tell whether they are chat terms within sentential context. In our experiments, 93 similar errors occurred. We believe this type of errors can be addressed within discoursal context.

Err.2 Chat terms created in manners other than phonetic mapping
Example-2: 忧虑 ing

In this example, XSCM does not recognize "ing" while the correct answer is "(正在)忧虑 (I'm worrying)". This is because chat terms created in manners other than phonetic mapping are excluded by the phonetic assumption in XSCM method. Around 1% chat terms fall out of phonetic mapping types. Besides chat terms holding same form as showed in Example-2, we find that emoticon is another major exception type. Fortunately, dictionary-based method is powerful enough to handle the exceptions. So, in a real system, the exceptions are handled by an extra component.

7 Conclusions

To address the sparse data problem and dynamic problem in Chinese chat text normalization, the phonetic mapping models are proposed in this paper to represent mappings between chat terms and standard words. Different from character mappings, the phonetic mappings are constructed from available standard Chinese corpus. We extend the source channel model by incorporating the phonetic mapping models. Three conclusions can be made according to our experiments. Firstly, XSCM outperforms SCM with same training data. Secondly, XSCM produces higher performance consistently on time-varying test sets. Thirdly, both SCM and XSCM perform best with biggest training chat language corpus.

Some questions remain open to us regarding optimal size of training chat language corpus in XSCM. Does the optimal size exist? Then what is it? These questions will be addressed in our future work. Moreover, bigger context will be considered in chat term normalization, discourse for instance.

Acknowledgement

Research described in this paper is partially supported by the Chinese University of Hong Kong under the Direct Grant Scheme project (2050330) and Strategic Grant Scheme project (4410001).

References

Brown, P. F., J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer and P. S. Roossin. 1990. A statistical approach to machine translation. Computational Linguistics, v.16 n.2, p.79-85.

Gianforte, G.. 2003. From Call Center to Contact Center: How to Successfully Blend Phone, Email, Web and Chat to Deliver Great Service and Slash Costs. RightNow Technologies.

Graf, D., K. Chen, J.Kong and K. Maeda. 2005. Chinese Gigaword Second Edition. LDC Catalog Number LDC2005T14.

Heard-White, M., Gunter Saunders and Anita Pincas. 2004. Report into the use of CHAT in education. Final report for project of Effective use of CHAT in Online Learning, Institute of Education, University of London.

James, F.. 2000. Modified Kneser-Ney Smoothing of n-gram Models. RIACS Technical Report 00.07.

Katz, S. M.. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing, 35(3):400-401.

Li, H., W. He and B. Yuan. 2003. An Kind of Chinese Text Strings' Similarity and its Application in Speech Recognition. Journal of Chinese Information Processing, 2003 Vol.17 No.1 P.60-64.

McCullagh, D.. 2004. Security officials to spy on chat rooms. News provided by CNET Networks. November 24, 2004.

Xia, Y., K.-F. Wong and W. Gao. 2005. NIL is not Nothing: Recognition of Chinese Network Informal Language Expressions. 4th SIGHAN Workshop at IJCNLP'05, pp.95-102.

Xia, Y. and K.-F. Wong. 2006a. Anomaly Detecting within Dynamic Chinese Chat Text. EACL'06 NEW TEXT workshop, pp.48-55.

Xia, Y., K.-F. Wong and W. Li. 2006b. Constructing A Chinese Chat Text Corpus with A Two-Stage Incremental Annotation Approach. LREC'06.