Lang Resources & Evaluation
DOI 10.1007/s10579-008-9067-7

Normalization of Chinese chat language

Kam-Fai Wong · Yunqing Xia

Abstract Real-time communication platforms such as ICQ, MSN and online chat rooms are getting more popular than ever on the Internet. There are, however, real risks where criminals and terrorists can perpetrate illegal and criminal abuses. This highlights the security significance of accurate detection and translation of the chat language to its stand language counterpart. The language used on these platforms differs significantly from the standard language. This language, referred to as chat language, is comparatively informal, anomalous and dynamic. Such features render conventional language resources such as dictionaries, and processing tools such as parsers ineffective. In this paper, we present the NIL corpus, a chat language text collection annotated to facilitate training and testing of chat language processing algorithms. We analyse the NIL corpus to study the linguistic characteristics and contextual behaviour of a chat language. First we observe that majority of the chat terms, i.e. informal words in a chat text, is formed by phonetic mapping. We then propose the eXtended Source Channel Model (XSCM) for the normalization of the chat language, which is a process to convert messages expressed in a chat language to its standard language counterpart. Experimental results indicate that the performance of XSCM in terms of chat term recognition and normalization accuracy is superior to its Source Channel Model (SCM) counterparts, and is also more consistent over time.

This is an extension of the paper presented at COLING/ACL 2006 (Xia et al. 2006b).

K.-F. Wong
Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, NT, Hong Kong
e-mail: kfwong@se.cuhk.edu.hk

Y. Xia (✉)
Centre for Speech and Language Technologies, RIIT, Tsinghua University, Beijing 100084, China
e-mail: yqxia@tsinghua.edu.cn

Keywords Chinese chat language · Phonetic mapping · Chat language Modelling · Chat term normalization · Natural language processing

1 Introduction

The Internet supports online chatting via different real-time communication platforms such as ICQ, MSN, online chat rooms, BBS, email, blogs, etc. The language used on these platforms, referred to as a chat language, is gaining global popularity, especially within the Chinese communities worldwide. Investigation reveals that chat language texts appear frequently in chat logs of online education (Heard-White et al. 2004), customer relationship management (CRM) (Gianforte 2003), and so on. Online chatting now become has now become an important communication medium making daily life easier. But, at the same time, many Internet chat rooms and BBS systems are constantly abused by perpetrators of terrorism, pornography and crime (McCullagh 2004). In general, whether it is to provide better services in online education or CRM, or to monitor and control the Web against terrorism, there is an urgent social and business need to understand the communications over the Internet. This highlights the significance of our project, the normalization of the chat language in the Chinese domain. Normalization implies translation of an expression in the chat language to its standard language counterpart.

We observe that, compared with the standard language, a chat language is anomalous and dynamic in nature (see Sect. 2). Words specific to the chat language, referred to as chat terms, are different from the standard language in two obvious ways: either they fall outside the standard language's vocabulary; or they represent different meanings from their standard language counterparts. We find that on average one chat term is found in every 2.74 sentences in a BBS text collection. Chat terms are not covered by conventional linguistic tools like dictionaries and thesauruses rendering existing natural language processing techniques ineffective. The dynamic nature of the chat language is attributed to the regular changes of chat terms and its grammatical structure. Since the chat language is an informal language (i.e. a dialect), users are free to make up their own words and phrase structures, which are mostly short; as such, the users could make better use of the communication bandwidth. Also, influenced by the contemporary social environment such as catchy phrases on television, chat terms and phrases are often creative and fashionable. For example, many popular chat terms used a year ago have been discarded and many new chat terms are created in the present year.

The anomalous nature of Chinese chat language was investigated in an earlier work (Xia et al. 2005). Pattern matching and support vector machines (SVM) methods were proposed to recognize ambiguous chat terms from BBS chat text. Experiments show that chat term recognition rate measured in f-1 reached 87.1%. It was, however, found that quality of both methods dropped significantly when the training set became older. The dynamic nature of Chinese chat language was investigated in another study (Xia and Wong 2006). An error-driven approach was

proposed to detect chat terms in Chinese chat texts. It made use of linguistic information embedded in both a standard Chinese corpus and the Network Informal Language (NIL) 1.0 corpus (Xia et al. 2006a). The standard Chinese corpus provided negative text samples and NIL 1.0 corpus positive ones. The approach worked out the confidence and entropy values of the input BBS text and used a threshold value to identify the embedded chat terms. Compare with existing methods, the proposed method performed consistently over a time-varying test set. However, the issue of chat term normalization was not addressed in that study (Xia et al. 2006a). Conventionally, a check in the dictionary is the simplest approach to handle term translation. But this is inapplicable to chat term normalization due to the serious Out-Of-Vocabulary (OOV) problem caused by the regularly changing nature of the chat language. The targets of this project are (1) to compile a sizeable chat language corpus; (2) to analyse the corpus to study the characteristics of the chat language; and (3) to design a practical chat term normalization algorithm based on the analysis.

The rest of this paper is organized as follows. In Sect. 2, a study of the linguistic characteristics and contextual behaviour of the chat language is presented. We present a character mapping-based source channel model method for chat term normalization in Sect. 3 and show its limitations and ineffective points. In Sect. 4, we introduce the concept of phonetic mapping and outline the technical details of the phonetic mapping model. Section 5 presents an extended source channel model, which incorporates the phonetic mapping models, for chat term normalization. Experimental results and error analysis are presented in Sect. 6. Finally, Sect. 7 draws preliminary conclusions.

2 Chat language feature analysis

2.1 A Chinese chat language corpus

The NIL corpus is a collection of Chinese chat language sentences compiled to facilitate training and testing of the chat language knowledge engineering tools and processing methods. The NIL corpus is constructed by manual annotation.

Sources of on-line chat language texts were not easy to find. Obtaining online chat logs maintained by ICQ, MSN and online chat rooms was complicated. This was mainly due to data privacy restriction. We therefore resolved to sources that are publicly accessible such as BBS discussion postings. We found that BBS text within "大嘴区 (meaning: free chat zone; Chinese pinyin: *da4 zui3 qu1*)" discussion zone in YESKY BBS system (http://www.bbs.yesky.com/bbs/) closely resembled the characteristics of Chinese chat language and contained a vast amount of chat terms. Thus, BBS postings in that zone were finally used as the text source to produce the NIL corpus.

An early version of the corpus, i.e. NIL 1.0, covered chat language text created between December 2004 and February 2005. In the current version, NIL 2.0, we included chat language text created from March 2005 to February 2006. 30,392 chat

Table 1 Distribution of chat terms in the two anomalous types

Anomalous type	No. of unique terms	No. of occur.	Percentage of occur.
Anomalous entry	650	20585	46.82
Anomalous sense	211	23378	53.18
Total	861	43963	100

sentences selected from 120,429 BBS postings were annotated in the NIL 2.0 corpus.[1] Within it, 861 chat terms occur 43,963 times.

We begin our corpus analysis by investigating the linguistic characteristics of the Chinese chat language, including anomalous types, ambiguous status, morphological forms and phonetic behaviour. Problems caused by these linguistic characteristics are also studied. This is followed by a study on the contextual behaviour of the chat language. The goal is to understand its dynamism and the associated problems, which in turn will provide useful insights for us to design the ultimate chat term normalization algorithm.

2.2 Linguistic characteristics

2.2.1 Anomalous types

A chat language is linguistically anomalous to its standard language. There are mainly two types of anomaly. Firstly, some chat terms do not exist in conventional dictionaries. They are referred to as anomalous entries. 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 Chinese chat language. Secondly, while some chat terms can be found in conventional dictionaries as standard entries, their meanings are unknown to the dictionaries, which are referred to as anomalous senses. For example, in chat text "偶 (even; *ou3*)" is often used to replace "我 (me, *wo3*)". But "偶 (even; *ou3*)" merely means "even" in a conventional dictionary. This substitution takes place as the two words sound similar in Chinese. Distribution of chat terms in the two anomalous types is presented in Table 1.

Table 1 reveals that 46.82% chat terms are anomalous entries to conventional dictionary and the remaining 53.18% chat terms uses anomalous senses. To handle chat terms in both cases, a chat language dictionary was suggested to collect all chat terms and their senses. This approach is, however, impractical as unknown chat terms are created and relinquished too frequently for any conventional dictionary update mechanism. Thus, we conclude that knowledge-based methods are ineffective for chat language processing.

[1] Unless stated otherwise, both NIL corpus and chat language corpus refer to NIL corpus 2.0 hereafter.

2.2.2 Ambiguous usages

In chat text, chat terms and standard words are inter-mixed. Thus, recognition of chat terms is not straightforward, in particular the case in anomalous sense (see Table 1). Further, like standard language terms, many chat terms have multiple senses and the actual meaning is context dependent. Table 2 shows the statistics of sense distribution in the NIL corpus.

Table 2 shows that 66.99% occurrences of the 861 chat terms are ambiguously used (i.e. more than one sense). Disambiguation complexity increases with the number of sense of a word. Words with two, three and four meanings are most significant. They occupy 51.40%, 6.95% and 5.20%, respectively. In the ambiguous chat terms, "偶 (even; *ou3*)" occurs most (i.e. 8,735 times) which is used to replace "我 (I; *wo3*)"; "JJ" appears second most (i.e. 5,405 times) which represents "姐姐 (older sister; *jie3 jie3*)". But surprisingly, 325 chat terms appear only once in the NIL corpus. This implies a serious data sparseness problem for statistical chat language processing (see also Sect. 2.3).

We grouped the ambiguous chat terms into the two aforementioned anomalous types in Table 3. Chat terms in the *anomalous sense* group are highly ambiguous as they present at least two senses, i.e. one in the chat language and the other in the standard language contexts. This group also reflects that chat terms and standard language words are well mixed in the NIL corpus and recognizing chat terms from the mixed text is an essential step in the normalization process. Chat terms in the *anomalous entry* group represent the chat language specific words. 29.38% of them are ambiguous (i.e., representing more than one sense). This accounts for a significant portion and should not be ignored in any language processing tools.

2.2.3 Morphological forms

Chat terms are composed in various forms. The majority of chat terms is formed by a string of Chinese characters. Many others are composed by strings of letters, numbers and a mixture of Chinese and alphanumeric characters. Table 4 presents the chat term composition distribution in the NIL corpus.

Table 2 Statistics on sense distribution among chat terms

No. of senses	No. of unique terms	No. of occur.	Percentage of occur.	Max. occur.	Avg. occur.
1	613	14514	33.01	5405	24
2	211	22598	51.40	8735	107
3	21	3054	6.95	502	145
4	13	2285	5.20	909	176
5	1	24	0.05	24	24
6	1	413	0.94	413	413
7	1	1075	2.45	1075	1075
Total	861	43963	100	–	–

Table 3 Distribution of ambiguous chat terms in the two anomalous types

Anomalous type	No. of senses	No. of unique terms	No. of occur.	Percentage of occur.	Overall %
Anomalous entry	1	614	14538	70.62	
	2	26	3975	19.31	
	3	2	782	3.80	
	4	8	1290	6.27	
	5	0	0	0.00	
	6	0	0	0.00	
	7	0	0	0.00	
	Total	650	20585	100	46.82
Anomalous sense	1	0	0	0.00	
	2	185	18623	79.66	
	3	19	2272	9.72	
	4	4	971	4.15	
	5	1	24	0.10	
	6	1	413	1.77	
	7	1	1075	4.60	
	Total	211	23378	100	53.18
Total		861	43963	–	100

Table 4 Different morphological forms in chat terms

Morphological form	No. of unique terms	No. of occur.	Percentage of occur.
Chinese character/ word/phrase	446	24438	55.59
Letter	197	15514	35.29
Number	25	2886	6.56
Mixed form	147	1029	2.34
Other	46	96	0.22
Total	861	43963	100

In conventional Chinese text, very few letters are used; but this is not the case in Chinese chat text. This is due mainly to the fact that letters can help reduce the burden of inputting Chinese characters to the computer. In the NIL corpus, 35.29% of the chat terms (i.e. 197 unique entries) consist of letters. It is noteworthy that in this category, 173 chat terms use Chinese pinyin abbreviations or initials, e.g. "JJ" representing "姐姐" and the rest uses English pronunciation or initials, e.g. ASAP.

It is common to find numbers in the Chinese standard language. But in the chat language, numeric characters do not always represent numbers. For example, "7 (seven; *qi1*)" is used in the chat sentence "我爱 7 牛肉 (I like to eat beef; *wo3 ai4 qi1 niu3 rou4*)" to represent "吃 (eat; *chi1*)".

The mixed form makes the chat language most different from the standard language. Chat terms in this type combine Chinese characters, letters and/or

numeric characters. The mixed form would present problems to conventional morphological analysis tools. Conventional Chinese word segmentation tools separate numbers and letters from Chinese characters. For example, "8错 (not bad; *ba1 cuo4*)" would be split to "8 (eight; *ba1*)" and "错 (wrong; *cuo4*)" by ICTCLAS (Zhang et al 2003), a popular word segmentation tool. But in fact it should be treated as one 'word' in the chat language representing "不错 (not bad; *bu3 cuo4*)".

There are 0.22% chat terms with other forms. They are mainly emotions, which make uses of combination of punctuations, numbers and letters to represent different emotions, e.g. ":-)" represents a happy face and ":-(" a sad face.

2.2.4 *Phonetic behaviour*

Our observation on phonetic behaviour of chat terms indicates that most chat terms are created using phonetic mappings instead of character mappings. In other words, most chat terms and their standard language counterparts are similar in phonetic transcription. For example, ignoring their tones, chat term "滴 (drop; *di1*)" and "地 (ground; *di4*)" share the same Chinese pinyin, i.e. *di*. In addition, formation of many Chinese chat terms is based on Chinese dialects rather than standard Chinese, i.e. Mandarin. For example, the chat term "粉 (powder; *fen3*)" and "很 (very; *hen3*)" are phonetically equal in a southern Chinese dialect. Table 5 presents distribution of chat terms in terms of phonetic behaviour.

Table 5 shows that 97.28% chat terms are formed based on phonetic mapping. This observation provides very important clues to chat language modelling and normalization. Intuitively, one would consider using a character mapping method to translate chat terms to their counterparts. However, this method would seriously suffer from data sparseness problem because a large chat language corpus is not available. What is even worse is that chat terms are created and relinquished quickly rendering available character mappings invalid to model them.

In contrast, the phonetic mapping method for chat term normalization is more flexible. Firstly, phonetic mappings can be produced beforehand using standard language corpus. This can ensure completeness of the phonetic mapping model. Secondly, the mapping space between chat terms and standard language words is significantly reduced by the phonetic mapping method. Chinese characters are first grouped and then mapped between each other via similar pinyin. Thus, phonetic mapping is actually based on group-to-group mapping. Our text corpus comprises of 5,095 simplified Chinese characters but only 735 pinyin units leading to a

Table 5 Chat term distribution in terms of phonetic behaviour

Phonetic behaviour	No. of unique terms	No. of occur.	Percentage of occur.
Created using phonetic clue	802	42767	97.28
Created using no phonetic clue	59	1196	2.72
Total	861	43963	100

significant reduction in the mapping space, i.e. around 7 times. Ambiguity arising from the phonetic grouping is inevitable, yet pinyin similarity and character frequency are found effective parameters for disambiguation in this research.

2.3 Contextual behaviour of chat terms

We define contextual behaviour of chat terms as the rates chat terms are created and relinquished. This reflects the dynamism of the chat language. We show in this section that the chat language is dynamic and that leads to serious sparse data problems.

2.3.1 Creation and relinquish rates

The chat language is dynamic. New terms are created and old terms are relinquished regularly. We define the creation rate as percentage that new chat terms are created and the relinquish rate the percentage that old chat terms are relinquished in 2 months. Suppose we have two chat term sets, i.e. TS_1 and TS_2, in 2 month periods T_1 and T_2 respectively where T_2 is later than T_1 and hence, TS_2 is newer than TS_1. The creation and relinquish rates are defined as follows:

$$rate\_creation(T_1, T_2) = \frac{|TS_1 \cap TS_2|}{|TS_2|} \tag{1}$$

$$rate\_relinqulish(T_1, T_2) = \frac{|TS_1 \cap TS_2|}{|TS_1|} \tag{2}$$

We group chat terms in batches of 2 months in the period from December 2004 to February 2006. Then their creation and relinquish rates are calculated using Eqs. 1 and 2 and presented in Tables 6 and 7, respectively.

Tables 6 and 7 reveal that, within 12 months (i.e. from Dec-04 to Dec-05), 17.28% chat terms are created and 17.82% chat terms relinquished. Such rates are different from the standard language, which changes slightly in more than 5 years.

Table 6 Chat term creation rates

Set	Feb-05	Apr-05	Jun-05	Aug-05	Oct-05	Dec-05	Feb-06
Dec-04	0.0231	0.0531	0.0912	0.1410	0.1640	0.1728	0.1880
Feb-05	–	0.0307	0.0697	0.1207	0.1442	0.1532	0.1688
Apr-05	–	–	0.0402	0.0928	0.1171	0.1264	0.1425
Jun-06	–	–	–	0.0548	0.0801	0.0898	0.1065
Aug-05	–	–	–	–	0.0268	0.0370	0.0547
Oct-05	–	–	–	–	–	0.0105	0.0287
Dec-05	–	–	–	–	–	–	0.0184

Table 7 Chat term relinquish rates

Set	Feb-05	Apr-05	Jun-05	Aug-05	Oct-05	Dec-05	Feb-06
Dec-04	0.0259	0.0560	0.0920	0.1265	0.1578	0.1782	0.1878
Feb-05	–	0.0309	0.0679	0.1033	0.1354	0.1563	0.1662
Apr-05	–	–	0.0381	0.0747	0.1078	0.1294	0.1396
Jun-06	–	–	–	0.0380	0.0725	0.0949	0.1055
Aug-05	–	–	–	–	0.0358	0.0592	0.0702
Oct-05	–	–	–	–	–	0.0242	0.0356
Dec-05	–	–	–	–	–	–	0.0117

Some event-related chat terms are convincing evidences for the above observation. It is observed that some chat terms were frequently used for only a short period of time. This happens especially in popular events. For example, the chat term "玉米 (corn; *yu4 mi3*)" appeared during the hottest Chinese TV show "超级女声 (Super Girl Voice; *chao1 ji2 nv3 sheng1*)" in 2006; and it replaced "宇迷" (fans of Yuchun, the Super Girl champion; *yu3 mi2*). Today, this chat term has become obsolete as that TV show was over. In general, since such terms are formed based on phonetic clues, they have been catered for in our research work.

2.3.2 Sparse data

This is a classical problem in statistical NLP approaches. It occurs when training data in the specific language domain are insufficient. Now we study the chat term distribution in terms of their occurrences (see Fig. 1).

We observe that in the NIL corpus, 540 out of the 861 unique chat terms occur less than five times. This would lead to serious sparse data problem in statistical learning. The problem is made even worse by the dynamic nature of the chat language. As shown in Tables 6 and 7, 17.28% chat terms are created and 17.82% relinquished in the first year. This would lead to further data sparseness since chat

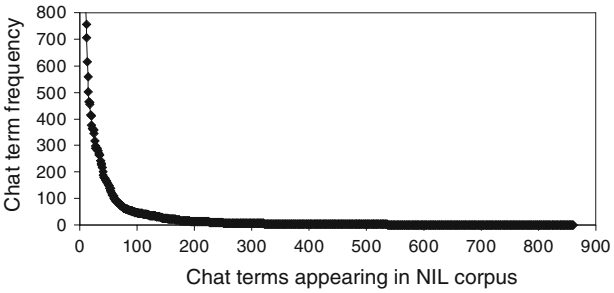

Fig. 1 Distribution of chat terms regarding number of occurrences. Axis *x* represents chat term and axis *y* number of chat term occurrences in NIL corpus. For demonstration purposes, the chat terms are sorted by number of their occurrences

language models trained on chat text collected last year would be outdated and hence, ineffective for processing chat texts this year.

3 Normalization with source channel model

In this paper we propose a new chat language modelling technique to address the problems caused by the anomalous and dynamic nature of the chat language. The goal of this research is to design an effective method to recognize chat terms in random chat language text and translate them to their standard language counterparts, i.e. normalization. In this section, the baseline method implemented using the classical source channel model (SCM) is outlined. We study the deficiency of SCM for chat term normalization and propose the extended source channel model (XSCM) in Sect. 5. We also use performance of SCM as the reference in evaluating the performance of XSCM in Sect. 6.

3.1 The source channel model

The source channel model (SCM) is a widely used statistical approach in speech recognition and machine translation (Brown et al. 1990). Since chat term normalization is very similar to these applications, SCM is deemed most appropriate for the task. Given an input chat text string, $T = \{t_i\}_{j=1,2,\ldots,n}$ where t_i's are input characters, SCM aims to find the most probable translation character string $C = \{c_i\}_{j=1,2,\ldots,n}$ where c_i's are the output characters, viz:

$$C = \arg\max_C p(C|T) = \arg\max_C p(T|C)p(C) \tag{3}$$

$p(C|T)$ comprises of two components: chat term translation model, i.e. $p(T|C)$ and language model, i.e. $p(C)$. $p(T|C)$ is actually a character mapping model produced with a chat language corpus. The two models can be estimated with the maximum likelihood method using the character trigram in the NIL corpus.

3.2 The problems

Two problems are worth noting in applying SCM in chat term normalization. First, data sparseness problem is serious because size of the timely chat language corpus is too small to provide sufficient character mappings. The NIL corpus contains only 30,292 chat sentences created in 15 months. This is insufficient to train the chat term translation observation model. Second, training effectiveness is poor again due to the dynamic nature of the chat language. Trained on static chat text pieces, the SCM approach would perform poorly in processing future chat text.

Updating the NIL corpus with recent chat text constantly is an ineffective solution to the above problems. It is desirable to find some linguistic information underlying chat terms to help address the data sparseness and dynamic problems.

Observations outlined in Sect. 2.2.4 provide evidence that phonetic mappings exist between most chat terms and their standard language counterparts. Thus, we apply such mappings in resolving the two problems.

4 Phonetic mapping models

4.1 Phonetic mapping

Phonetic mapping connects two characters via phonetic transcription, i.e. Chinese pinyin in our case. For example,

$$介 \xrightarrow{(zhe,\ jie,\ 0.56)} 这$$

is a phonetic mapping connecting "这 (this; *zhe4*)" and "介 (interrupt; *jie4*)", in which "*zhe*" and "*jie*" are Chinese pinyin for "这" and "介", respectively. The number 0.56 in the bracket is phonetic mapping probability between the two characters. Some categorized examples of phonetic mappings are given in Appendix 1.

Technically, phonetic mappings can be constructed between any two characters within any Chinese corpus. In the chat language, as any Chinese characters can be used in a chat term, phonetic mappings are used to translate the chat term to its standard language counterpart. Different from character mappings which are merely extracted from the chat language corpus, phonetic mappings can be obtained from a standard Chinese corpus; and the chat language corpus is then used to refine the phonetic mapping probabilities.

4.2 Phonetic assumption and justifications

To make use of phonetic mappings as the fundamental knowledge in chat term normalization, the following phonetic assumption is made.

Phonetic assumption: In the Chinese chat language, chat terms are mainly created using phonetic mappings and the phonetic mappings are stable over time.

To ensure that the assumption holds in our method, two questions must be answered. First, how many percentage of chat terms are created via phonetic mappings? Second, why are phonetic mappings stable and character mappings not in the chat language? The first question has already been answered in phonetic behaviour analysis on the chat language in Sect. 2.2.4. We would like to focus on the second question.

Analysis on creation/relinquish rates in Sect. 2.3 shows that chat terms evolve dynamically. The analysis, however, examined character rather than phonetic behaviour. We conducted another analysis on fifteen chat term sets, i.e. one for each month of the NIL corpus from December 04 to February 06 (see Sect. 2.1) to investigate how phonetic mappings of these chat terms behave over time. We created fifteen chat language phonetic mapping sets, one for each of the fifteen chat term sets and a standard phonetic mapping set using Chinese Gigaword Second

Edition (Graf et al. 2005). We compared each of the fifteen phonetic mapping sets against the standard set and observed that the standard set consistently covered more than 97% phonetic mappings in each month. Compare with the creation/relinquish rates in Sect. 2.3, we are convinced that the phonetic mappings constructed with the standard Chinese language are more stable over time.

4.3 Formalism

Phonetic mapping is modelled by a five-element tuple, i.e.

$$\langle t, c, pt(t), pt(c), \Pr_{pm}(t|c) \rangle$$

which comprises of an input chat term character t; the output standard language counterpart, character c; phonetic transcriptions of t and c, i.e. $pt(t)$ and $pt(c)$; and the mapping probability $\Pr_{pm}(t|c)$ in which t is mapped to c via the phonetic mapping:

$$t \xrightarrow{(pt(t),\ pt(c),\ \Pr_{pm}(t|c))} c \text{ (briefed by } t \xrightarrow{m} c \text{ hereafter).}$$

Since phonetic mappings concern mappings between any Chinese character pairs via pinyin and the characters are not necessarily related to chat terms, they could be obtained from a standard language corpus. This results in two advantages: (1) the impact of sparse data problem is reduced as the standard language corpus can provide broader coverage (see Sect. 4.2); and (2) the phonetic mapping model is as stable as the standard language. As such, in chat term normalization, when the phonetic mapping models are used to represent mappings between chat term characters and their standard language counterparts, the dynamic problem can be addressed effectively. In contrast, SCM adopts the character mapping model (see Sect. 3.1). The model connects two Chinese characters directly. It is modelled by a three-element tuple, i.e.

$$\langle t, c, \Pr_{cm}(t|c) \rangle,$$

which comprises the input chat term character t, the corresponding output standard language character c and the character mapping probability $\Pr_{cm}(t|c)$ that t is mapped to c via this character mapping. As they must be constructed from the chat language training set, which is significantly smaller than the standard language corpus, it is very likely that the character mapping model suffers more from both the data sparseness and dynamic problems.

4.4 Parameter estimation

To construct the phonetic mapping models, we first extract all Chinese characters from a standard Chinese language corpus and use them to form the character mapping models. We then generate phonetic transcriptions of the Chinese

characters and calculate the phonetic mapping probability for each character mapping. We exclude those character mappings holding zero probability. Finally, character mappings are converted to phonetic mappings by phonetic transcription and the phonetic mapping probability of each conversion is incorporated. Specifically, this is how the phonetic mapping probabilities are estimated.

The phonetic mapping probability is calculated by combining phonetic similarity and character frequency in the standard language as

$$\Pr_{pm}(c|\bar{c}) = \frac{(fr_{slc}(\bar{c}) \times ps(c,\bar{c}))}{\sum_i (fr_{slc}(c_i) \times ps(c,c_i))} \tag{4}$$

where $\{c_i\}$ is the character set in which each element c_i is similar to character c in terms of phonetic transcription. $fr_{slc}(x)$ is a function that returns frequency of character x in the standard language corpus and $ps(x_1,x_2)$ returns phonetic similarity between characters x_1 and x_2.

Phonetic similarity between two Chinese characters is calculated based on Chinese pinyin as

$$\begin{aligned} ps(c,c) = Sim(py(c),py(c)) = Sim(initial(py(c)),initial(py(c))) \\ \times Sim(final(py(c)),final(py(c))). \end{aligned} \tag{5}$$

where $py(x)$ is a function that returns Chinese pinyin of character x, and *initial*(y) and *final*(y) return initial (*shengmu*) and final (*yunmu*) of Chinese pinyin y, respectively. For example, pinyin for Chinese character "这 (this; *zhe4*)" is "*zhe*", in which "*zh*" is the pinyin initial and "*e*" the pinyin final. In cases where either an initial or a final is empty, we use similarity of the existing parts, e.g. we calculate phonetic similarity of "撒 (scatter; *sa3*)" and "啊 (ah, an exclamation; *a4*)" as follows.

$$Sim(a,sa) = Sim(final(a),final(sa)) = Sim(a,a)$$

An algorithm to calculate the similarity of initial pairs and final pairs is proposed in (Li et al. 2003) based on letter matching. The problem of this algorithm is that it always assigns zero similarity to those pairs containing no common letter. For example, the initial similarity between "*ch*" and "*q*" is set to zero by this algorithm. However, in fact, pronunciations of the two initials are very close to each other in spoken Chinese. For this reason, non-zero similarity values should be assigned to these special pairs beforehand (e.g., similarity between "*ch*" and "*q*" is set to 0.8). All similarity values have been validated by several native Chinese speakers. Thus the aforementioned algorithm is extended to output a pre-defined similarity value before letter matching. For example, pinyin similarity between "*chi*" and "*qi*" is calculated as follows.

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

At this point, $\Pr_{pm}(c|\bar{c})$ is only estimated with standard language corpus. We further propose to tune $\Pr_{pm}(c|\bar{c})$ using character frequencies in the NIL corpus. $\Pr_{pm}(c|\bar{c})$ is then rewritten as

$$\Pr^*_{pm}(c|\bar{c}) = \frac{fr_{NIL}(\bar{c}) \times \Pr_{pm}(c|\bar{c})}{\sum_i fr_{NIL}(c_i) \times \Pr_{pm}(c|c_i)} \tag{6}$$

where $fr_{NIL}(x)$ returns frequency of character x in the NIL corpus. As some character might not appear in the NIL corpus, we choose to assign a smoothing frequency to each of those zero-frequency characters based on its frequency in the standard language corpus, i.e.

$$sf(c_i) = \frac{fr_{slc}(c_i)}{\sum_j fr_{slc}(c_j)} \tag{7}$$

Equation 6 is then rewritten as

$$\Pr^*_{pm}(c|\bar{c}) = \frac{fr'_{NIL}(\bar{c}) \times \Pr_{pm}(c|\bar{c})}{\sum_i fr'_{NIL}(c_i) \times \Pr_{pm}(c|c_i)} \tag{8}$$

where $fr'_{NIL}(x)$ returns $fr_{NIL}(x)$ if character x appear in the NIL corpus and $sf(x)$ otherwise.

5 The extended source channel model

To handle the problems encountered in the method based on source channel models, we propose to extend the source channel model by inserting a phonetic mapping model $M = \{m_i\}_{i=1,2,\ldots,n}$ into Eq. 3, in which chat term character t_i is mapped to standard character c_i via phonetic mapping m_i, i.e. $t_i \xrightarrow{m_i} c_i$. The extended source channel model (XSCM) is formulated as follows.

$$\hat{C} \approx \arg\max_{M,C} p(T, M|C)p(C) = \arg\max_{M,C} p(T|M, C)p(T|M, C)p(M|C)p(C) \tag{9}$$

Three components are involved in XSCM, i.e. chat term normalization observation model $p(T|M,C)$, phonetic mapping model $p(M|C)$ and language model $p(C)$.

The chat term normalization model: We assume that phonetic mappings between Chinese chat terms and their standard language counterparts are independent of each other. Thus chat term normalization probability can be derived as follows.

$$p(T|M, C) = \prod_i p(t_i|m_i, c_i) \tag{10}$$

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

The phonetic mapping model: We assume that the phonetic mapping model depends merely on the current observation and calculate the phonetic mapping probability as follows.

$$p(M|C) = \prod_i p(m_i|c_i) = \prod_i \Pr^*_{pm}(t_i|c_i) \tag{11}$$

in which $\Pr^*_{pm}(t_i|c_i)$'s are estimated with Eqs. 4–8 on a standard Chinese language corpus.

The language model: The language model $p(C)$ can be estimated using maximum likelihood estimation method with Chinese character trigram model on the standard Chinese language corpus. In our implementation, the Katz Backoff smoothing technique (Katz 1987) is used to handle the sparse data problem and Viterbi algorithm (Manning and Schütze 1999) is employed to search for the optimal solution in XSCM.

6 Evaluation

6.1 Data sets

6.1.1 Training sets

Two types of training data are used in our experiments. We use the Chinese Gigaword Second Edition (CNGIGA) as the standard Chinese language corpus to construct phonetic mapping models because of its excellent coverage of the standard Simplified Chinese. We use the NIL 2.0 corpus as the chat language corpus (see Sect. 2.1).

To evaluate our method on time-varying training data, five chat language corpora, i.e. CT#1–CT#5, are created with NIL corpus (see Table 8). To evaluate our method on size-varying training data, the five time-varying training sets are in turn used to produce five size-varying training sets, i.e. CS#1–CS#5 (see Table 9). The size-varying training sets are created by accumulating the time-varying training sets from recent set (i.e., CT#5) to remote set (i.e., CT#1). This treatment accords to the way that people expand training set for dynamic language in real applications. Basically, a recent text is more similar to a contemporary text than a remote text, thus it is more useful in expanding the training set for a dynamic language.

6.1.2 Test sets

We extracted 1,000 chat language sentences posted each month from January 2006 to June 2006 and compiled six time-varying test sets, T#1–T#6, in which timestamp of T#1 was the earliest and that of T#6 the newest. Notice that the NIL corpus covered chat sentences in January and February 2006. Since it overlapped with T#1

Table 8 Time-varying chat language training sets

Training set	No. of chat sentences	Months covered (3 months each)
CT#1	6127	12-2004 to 2-2005
CT#2	6060	3-2005 to 5-2005
CT#3	6089	6-2005 to 8-2005
CT#4	6046	9-2005 to 11-2005
CT#5	6070	12-2005 to 2-2006

Table 9 Size-varying chat language training sets

Training set	No. of chat sentences	Months covered
CS#1	6070	12-2005 to 2-2006 (recent 3 months)
CS#2	12116	9-2005 to 2-2006 (recent 6 months)
CS#3	18205	6-2005 to 2-2006 (recent 9 months)
CS#4	24265	3-2005 to 2-2006 (recent 12 months)
CS#5	30392	12-2004 to 2-2006 (all 15 months)

and T#2, these two test sets were regarded as closed test sets and the others open ones. For evaluation purpose, standard answers were produced manually from the six test sets.

6.2 Evaluation criteria

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

$$p = \frac{x}{x+y} \quad r = \frac{x}{x+z} \quad f = \frac{2 \times p \times r}{p+r} \tag{12}$$

where x denotes number of correctly recognized chat terms, y number of incorrectly recognized chat terms and z number of unrecognized chat terms.

For normalization, we used accuracy (a), which was commonly accepted by machine translation researchers as a standard evaluation criterion. The normalization accuracy is defined as percentage of correctly normalized chat terms in all chat terms in the test set. Every output of the normalization methods was compared to the standard answer to produce the corresponding normalization accuracy.

6.3 Experiment I: Time-varying chat language corpora

The objective of this experiment is to prove two claims. First, the chat language is dynamic. Second, XSCM is more effective to handle the dynamic problem.

In this experiment we used five time-varying chat language corpora, i.e. CT#1–CT#5, as the training chat language corpus, respectively and used six time-varying test sets, i.e. T#1–T#6, for testing. In each test, SCM and XSCM were trained on one time-varying chat language corpus and tested on the six time-varying test sets. Recognition f-1 measure (f) and normalization accuracy (a) are presented in Table 10.

Sets of f-1measure and accuracy curves are showed in Fig. 2. Figure 3 shows performance gap between SCM and XSCM. These curves reveal three tendencies in the experimental results.

Table 10 Results of SCM and XSCM using time-varying chat language training set only

Training set			T#1	T#2	T#3	T#4	T#5	T#6
SCM	CT#1	f	0.796	0.794	0.773	0.738	0.692	0.635
		a	0.797	0.794	0.777	0.747	0.706	0.661
	CT#2	f	0.807	0.804	0.783	0.750	0.706	0.651
		a	0.805	0.803	0.785	0.757	0.720	0.678
	CT#3	f	0.813	0.812	0.791	0.761	0.720	0.666
		a	0.810	0.808	0.791	0.765	0.733	0.691
	CT#4	f	0.823	0.821	0.798	0.769	0.729	0.683
		a	0.816	0.816	0.799	0.772	0.741	0.706
	CT#5	f	0.831	0.830	0.809	0.778	0.739	0.695
		a	0.826	0.823	0.806	0.781	0.750	0.716
XSCM	CT#1	f	0.841	0.839	0.824	0.802	0.778	0.751
		a	0.834	0.832	0.817	0.798	0.773	0.748
	CT#2	f	0.847	0.845	0.831	0.81	0.788	0.761
		a	0.840	0.837	0.823	0.804	0.782	0.757
	CT#3	f	0.851	0.850	0.836	0.816	0.794	0.77
		a	0.843	0.842	0.828	0.810	0.791	0.765
	CT#4	f	0.857	0.855	0.840	0.822	0.801	0.779
		a	0.848	0.845	0.831	0.815	0.795	0.773
	CT#5	f	0.859	0.859	0.845	0.828	0.809	0.789
		a	0.850	0.850	0.837	0.820	0.803	0.782

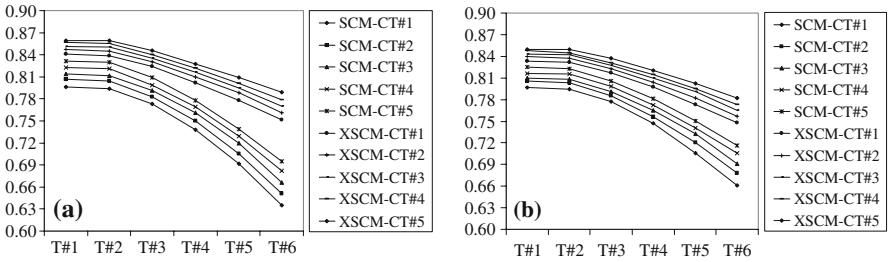

Fig. 2 Performance of SCM and XSCM using merely time-varying chat language corpus on six test sets. (**a**) Recognition f-1 measure; (**b**) normalization accuracy

i. Performance of both methods dropped on the same test sets when they were trained with the five time-varying chat language corpora. For example, both SCM and XSCM performed best on the newest training chat language corpus CT#5 and worst on the oldest corpus CT#1. This reflected the dynamic nature of the chat language.
ii. Performance of both methods dropped on the time-varying test sets under the same training chat language corpus. For example, both SCM and XSCM

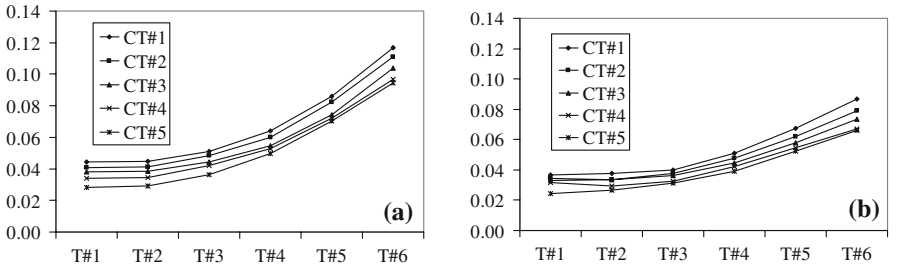

Fig. 3 Performance gap between SCM and XSCM on six test sets. (**a**) Recognition f-1 measure; (**b**) normalization accuracy

Table 11 Results of XSCM using time-varying chat language training set and CNGIGA

Training set			T#1	T#2	T#3	T#4	T#5	T#6
XSCM	CT#1	f	0.874	0.877	0.851	0.850	0.850	0.851
		a	0.877	0.878	0.850	0.848	0.850	0.848
	CT#2	f	0.875	0.875	0.852	0.851	0.852	0.851
		a	0.877	0.878	0.851	0.850	0.851	0.849
	CT#3	f	0.874	0.875	0.852	0.850	0.851	0.851
		a	0.878	0.879	0.852	0.851	0.851	0.850
	CT#4	f	0.876	0.877	0.853	0.852	0.853	0.851
		a	0.879	0.880	0.853	0.853	0.854	0.851
	CT#5	f	0.876	0.878	0.853	0.854	0.854	0.852
		a	0.880	0.881	0.854	0.854	0.855	0.853

performed best on the test set T#1 and worst on T#6. We found that the quality drop was caused by the dynamic nature of the chat language. It again revealed the dynamic nature of the chat language.

iii Performance gap between SCM and XSCM became bigger when the test set became newer. The gap was smallest on the oldest test set T#1 and biggest on the newest test set T#6. This showed the superiority of XSCM over SCM in dealing with the dynamic problem.

It should be pointed out that performance of XSCM dropped as the time-varying test sets became newer. This might be considered as counter-proof to our claim that XSCM could achieve high quality chat term normalization consistently. We found that this was due to insufficient training data. The NIL corpus was the only training corpus in this experiment and its coverage of phonetic mapping was limited. Thus, XSCM was in a sense under-trained leading to the performance drop. For this reason, we introduced the standard language corpus, i.e. CNGIGA, in XSCM training and re-ran the experiments. Recognition f-1 measure (*f*) and normalization accuracy (*a*) in the revised experimental are shown in Table 11.

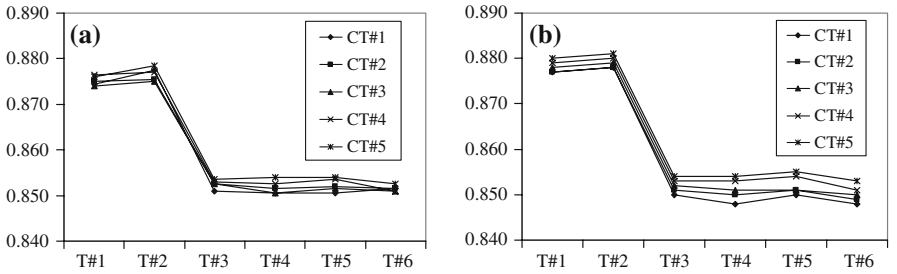

Fig. 4 Performance of XSCM using time-varying chat language corpus and CNGIGA on six test sets. (**a**) Recognition f-1 measure; (**b**) normalization accuracy

Compared with Tables 10, 11 presents the same values for SCM but much better values for XSCM. Two conclusions are drawn.

i. CNGIGA improved the performance of XSCM.
ii. CNGIGA facilitated XSCM to perform consistently well over all test sets. CNGIGA did not contribute any improvement in SCM because the standard corpus contains no annotation of chat terms. Contribution of CNGIGA to XSCM is shown clearly in Fig. 4.

However, the curves presented in Fig. 6 show that performance gain on both recognition f-1 measure (f) and normalization accuracy (a) saturates at CS#4. Accuracy gain on CS#5 over CS#4 is very little, i.e. around 0.001. We can thus conclude that size of corpus CS#4, i.e. 24,265, would be enough for XSCM to produce satisfactory performance and increasing training size beyond that number would not yield any noticeable performance gain.

6.4 Experiment II: size-varying chat language corpora

Although satisfactory performance was achieved in Experiment I, it was still uncertain whether the performance of XSCM could be further improved by increasing the size of the chat language corpus, i.e. the training corpus.

In this experiment, XSCM was trained on each of the five size-varying chat language corpora, i.e. CS#1–CS#5 and CNGIGA and tested on the six test sets T#1–T#6. Recognition f-1 measure (f) and normalization accuracy (a) are presented in Table 12.

Table 12 shows training size influences the performance of XSCM. XSCM performs best on the largest training chat language corpus, i.e. CS#5, and worst on the smallest, i.e. CS#1. Figure 5 reveals that XSCM favours bigger chat language corpus for training. Thus extending the chat language corpus should be one choice to improve the quality of chat language term normalization.

However, the curves presented in Fig. 6 show that the performance gain on both recognition f-1 measure (f) and normalization accuracy (a) saturates at CS#4.

Table 12 Results of XSCM using size-varying chat language training set and CNGIGA

Training set			T#1	T#2	T#3	T#4	T#5	T#6
XSCM	CS#1	f	0.876	0.878	0.853	0.854	0.854	0.852
		a	0.880	0.881	0.854	0.854	0.855	0.853
	CS#2	f	0.891	0.892	0.865	0.867	0.864	0.864
		a	0.893	0.893	0.868	0.870	0.865	0.864
	CS#3	f	0.900	0.901	0.873	0.876	0.874	0.874
		a	0.900	0.900	0.877	0.880	0.877	0.877
	CS#4	f	0.905	0.905	0.877	0.879	0.877	0.879
		a	0.903	0.903	0.880	0.884	0.881	0.881
	CS#5	f	0.906	0.907	0.879	0.880	0.879	0.880
		a	0.905	0.905	0.882	0.885	0.882	0.882

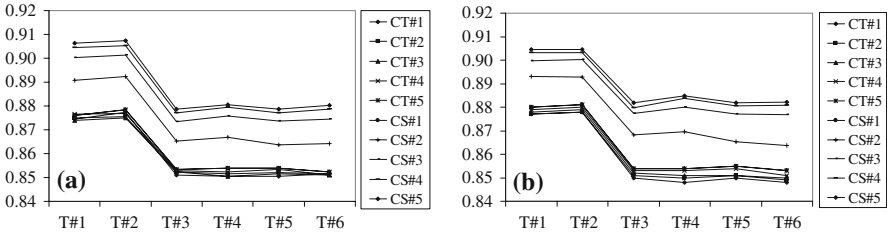

Fig. 5 Performance of XSCM using time-varying chat language corpus and size-varying chat language corpus on six test sets. (**a**) Recognition f-1 measure; (**b**) normalization accuracy

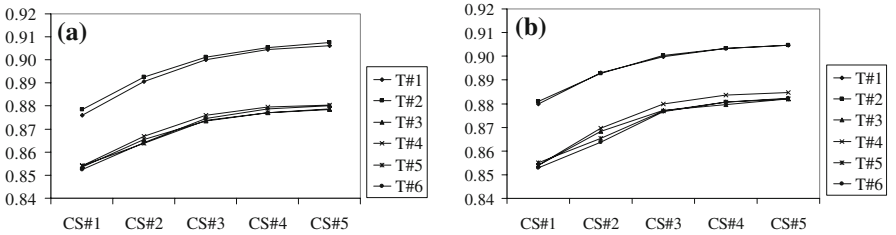

Fig. 6 Performance tendency of XSCM using size-varying chat language corpus on six test sets. (**a**) Recognition f-1 measure; (**b**) normalization accuracy

Accuracy gain on CS#5 over CS#4 is very little, i.e. around 0.001. We can thus conclude that the size of corpus CS#4, i.e. 24,265, would be enough for XSCM to produce satisfactory performance and increasing training size beyond that number would not yield any noticeable performance gain.

6.5 Error analysis

Table 12 shows that XSCM achieved 0.88 f-1 measure and 0.88 normalization accuracy in Experiment II. The 0.12 error rate is mainly caused by three types of errors.

Err. 1: Ambiguous chat terms.

Example-1: 他哈跑了 (He was scared away; *ta1 ha1 pao3 le1*)
Output: 他告跑了 (He ran away; *ta1 a4 pao3 le1*)
Answer: 他吓跑了 (He was scared away; *ta1 xia4 pao3 le1*)

In Example-1, "哈 (ha, an exclamation; *ha1*)" is a chat term representing "吓 (scare; *xia4*)". This error occurs because "哈" can be used to express an exclamation depicting laughing in standard Chinese language. We find "哈" can represent seven senses in the chat language. It is difficult for XSCM to disambiguate such a chat term with multiple senses. For example, "哈" can be normalized to "啊 (ah, an exclamation; *a1*)" (i.e., the output of XSCM in this case) because it proceeds a pronoun, i.e. "他 (he; *ta1*)". However, the chat term can be used to represent "吓" when a verb "跑 (run way; *pao3*)" follows. Unfortunately, "跑" was found more possible by XSCM to be the normalization counterpart for chat term "哈". In Experiment II, 197 errors of this type were caused by ambiguity.

Err. 2: Unrecognized chat terms.

Example-2: 我索八二年生 (I was born in 1982; *wo3 suo3 ba1 er4 nian3 sheng1*)
Output: 我索八二年生 (*The sentence doesn't make sense in standard Chinese*)
Answer: 我是八二年生 (I was born in 1982; *wo3 shi4 ba1 er4 nian3 sheng1*)

In Example-2, the chat term "索 (rope, *suo3*)" representing "是 (was, *shi4*)" is not recognized by XSCM. This is because phonetic similarity between "索" and "是" is too low (i.e., 2.1e-9) to be significant in the phonetic mappings. Eight errors of this type occurred in Experiment II. It is thus revealed that chat terms holding very low phonetic similarity might be mistakenly ignored by XSCM.

In Example-3, XSCM cannot recognize "粉丝 (vermicelli; *fen3 si1*). This is because the chat term is created using English-Chinese transliteration (Gao et al. 2004) instead of phonetic mapping between two Chinese terms. Although transliteration could be considered as another type of phonetic mapping, i.e. cross-lingual phonetic mapping, it is not catered for in our approach.

Err. 3: Chat terms created in manners other than phonetic mapping.

Example-3: 他们是粉丝 (They are fans; *ta1 men2 shi4 fen3 si1*)
Output: 他们是粉丝 (They are vermicelli; *ta1 men2 shi4 fen3 si1*)
Answer: 们是爱好者 (They are fans; *ta1 men2 shi4 ai4 hao4 zhe3*)

It is shown in Table 5 that around 2.72% chat terms did not contain any phonetic clue. They include English-Chinese transliteration (Gao et al. 2004) (e.g., "粉丝" represents the English word "fans"), multiple phonetic mappings (e.g., "表 (watch; *biao3*)", a short form phonetic representation of "不要 (do not; *bu3 yao4*)"),

emoticons (e.g., ":-)" represents "happy") and some idiosyncratic/personal usages (e.g., "9" represents monkey). 56 errors of this type occurred in Experiment II. In practice, we used a dictionary to handle these exceptions before we applied the phonetic mapping method.

7 Conclusions

We presented a Chinese chat language corpus, namely NIL corpus 2.0, which is the first text collection of this kind. NIL corpus is useful to research in chat language processing. Analysis of the NIL corpus reveals that the chat language is dynamic in nature and anomalous to the standard language rendering conventional NLP resources and tools ineffective. We also observed that most chat terms are similar to some forms of phonetic transcription of their standard language counterparts.

In addition, we offered an introduction to the normalization of chat terms, the process to translate a chat term to its standard language counterpart. Source channel model (SCM) is examined for this purpose, which is found ineffective as its translation model is based on character mapping. We extended SCM by incorporating the phonetic mapping model resulting in the XSCM method. XSCM trained with NIL corpus outperforms SCM under the same training condition in both chat term recognition and chat term normalization. Meanwhile, we demonstrated that by further training the XSCM with a standard Chinese language corpus (i.e. CNGIGA), its performance becomes more stable. However, there are around 12% errors in the existing implementation. They are mainly due to special chat term types. Contextual and semantic analysis techniques can be used to overcome them.

At present, we only focused on chat term normalization. However, full-fledged chat language normalization also involves sentences. Preliminary review shows that, compared with standard language sentences, chat sentences are shorter in length, often ungrammatical, anomalous in word order, and often with ellipsis. These are characteristics of human dialogue and it often involves multiple parties. These will form the major core of our continuous research in the "Chinese Chat Language Normalization".

Furthermore, it is worth noting that today many people are concerned with the impact of network terms on human languages (Cheng 2004, http://www.tech.163.com/special/w/wlyy.html). But the contextual behaviour of chat terms has never been studied systematically. In this paper, our research findings on the contextual behaviour of chat terms are presented in details. This helps linguists understand how chat terms are created and relinquished. This in turn will provide foundation for social linguistic research in Network Informal Languages (NIL). Observation on life cycles of new words for each language is an important research issue. Metcalf has been tracking English new words as they arise for 60 years (Metcalf 2002). Enlightened by his work, we plan to track life cycles of Chinese chat terms in the future to see how they gradually evolve to become standard words.

Acknowledgement Research described in this paper is partially supported by The Chinese University of Hong Kong under the Direct Grant Scheme project (No. 2050330 and 2050417), Strategic Grant Scheme project (No. 4410001) and NSFC (No. 60703051). We would also like to thank the reviewers for their valuable advices on this paper.

Appendix 1: Some categorized examples of phonetic mappings

1. Chinese to Chinese phonetic mappings
   (1) 偶 $\xrightarrow{(wo,ou,0.685)}$ 我: 偶 (even; *ou3*) replaces我 (me, *wo3*) with $p = 0.685$.
   (2) 介 $\xrightarrow{(zhe,jie,0.56)}$ 这: 介 (interrupt; *jie4*) replaces 这 (this; *zhe4*) with $p = 0.560$.
   (3) 素 $\xrightarrow{(shi,su,0.491)}$ 是: 素 (white, *su4*) replaces 是 (is, *shi4*) with $p = 0.491$.
   (4) 银 $\xrightarrow{(ren,yin,0.457)}$ 人: 银 (silver, *yin2*) replaces 人 (human, *ren2*) with $p = 0.457$.
   (5) 米 $\xrightarrow{(mei,mi,0.452)}$ 没: 米 (rice, mi3) replaces 没 (have not, *mei2*) with $p = 0.452$.
2. Letter to Chinese phonetic mappings
   (6) J $\xrightarrow{(jie,ji,0.671)}$ 姐: *J* replaces 姐 (older sister; *jie3*) with $p = 0.671$.
   (7) M $\xrightarrow{(mei,mi,0.593)}$ 妹: *M* replaces 妹 (younger sister; *mei4*) with $p = 0.593$.
   (8) S $\xrightarrow{(si,si,0.587)}$ 死: *S* replaces 死 (die; *si3*) with $p = 0.587$.
   (9) T $\xrightarrow{(ti,ti,0.465)}$ 踢: *T* replaces 踢 (kick; *ti1*) with $p = 0.465$.
   (10) K $\xrightarrow{(kuai,ki,0.447)}$ 快: *K* replaces 快 (quick; *kuai4*) with $p = 0.447$.
3. Number to Chinese phonetic mappings
   (11) 9 $\xrightarrow{(jiu,jiu,0.541)}$ 酒: 9 replaces 酒 (wine; *jiu3*) with $p = 0.541$.
   (12) 8 $\xrightarrow{(bu,ba,0.519)}$ 不: 8 replaces 不 (no; *bu4*) with $p = 0.519$.
   (13) 7 $\xrightarrow{(chi,qi,0.454)}$ 吃: 7 replaces 吃 (eat; *chi1*) with $p = 0.454$.
   (14) 4 $\xrightarrow{(si,si,0.449)}$ 死: 4 replaces 死 (die; *si3*) with $p = 0.449$.
   (15) 5 $\xrightarrow{(wu,wu,0.297)}$ 呜: 5 replaces 呜 (crying sound; *wu1*) with $p = 0.297$.

References

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics, 16*(2), 79–85.

Cheng, C. (2004). Network language: Advance or degeneration of Chinese language? http://www.tech.163.com/special/w/wlyy.html.

Gao, W., Wong, K.-F., & Lam, W. (2004). Phoneme-based transliteration of foreign names for OOV problem. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP'04)*, Sanya, China, 22–24 March, pp. 110–119.

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., Chen, K., Kong, J., & Maeda, K. (2005). *Chinese gigaword* (2nd ed.). LDC Catalog Number LDC2005T14.

Heard-White, M., Saunders, G., & Pincas, A. (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.

Katz, S. M. (1987). 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., He, W., & Yuan, B. (2003). A kind of Chinese text strings' similarity and its application in speech recognition. *Journal of Chinese Information Processing, 17*(1), 60–64.

Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

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

Metcalf, A. (2002). *Predicting new words: The secrets of their success*. Houghton Mifflin.

Xia, Y., & Wong, K.-F. (2006). Anomaly detecting within dynamic Chinese chat text. In *Proceedings of NEW TEXT Workshop at the 11th Conference for European Chapter of the Association for Computational Linguistics (EACL'06)*, Trento, Italy, 3–7 April, pp. 48–55.

Xia, Y., Wong, K.-F., & Gao, W. (2005). NIL is not nothing: Recognition of Chinese network informal language expressions. In *Proceedings of 4th SIGHAN Workshop at International Joint Conference on Natural Language Processing (IJCNLP'05)*, Jeju Island, Republic of Korea, 11–13 October, pp. 95–102.

Xia, Y., Wong, K.-F., & Li, W. (2006a) Constructing a Chinese chat text corpus with a two-stage incremental annotation approach. In *Proceedings of The 5th International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, 24–26 May.

Xia, Y., Wong, K.-F., & Li, W. (2006b). A phonetic based approach to Chinese chat term normalization. In *Proceedings of COLING/ACL Joint Conference*, Sydney, Australia, 17–21 July, Vol. 2, pp. 993–1000.

Zhang, Z., Yu, H., Xiong, D., & Liu, Q. (2003). HMM-based Chinese lexical analyzer ICTCLAS. In The 2nd SIGHAN Workshop Affiliated with ACL'2003, Sapporo, Japan, 11–12 July, pp. 184–187.