Constructing A Chinese Chat Language Corpus with A Two-Stage Incremental Annotation Approach

Yunqing Xia[1], Kam-Fai Wong[1] and Wenjie Li[2]

[1]Department of S.E.E.M., The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
Email: {yqxia, kfwong}@se.cuhk.edu.hk

[2]Department of Computing, The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: cswjli@comp.polyu.edu.hk

Abstract

Chat language refers to the special human language widely used in the community of digital network chat. As chat language holds anomalous characteristics in forming words, phrases, and non-alphabetical characters, conventional natural language processing tools are ineffective to handle chat language text. Previous research shows that knowledge based methods perform less effectively in processing unseen chat terms. This motivates us to construct a chat language corpus so that corpus-based techniques of chat language text processing can be developed and evaluated. However, creating the corpus merely by hand is difficult. One, this work is manpower consuming. Second, annotation inconsistency is serious. To minimize manpower and annotation inconsistency, a two-stage incremental annotation approach is proposed in this paper in constructing a chat language corpus. Experiments conducted in this paper show that the performance of corpus annotation can be improved greatly with this approach.

Keywords chat language, corpus annotation, natural language processing

1. Introduction

We refer network informal language (NIL) to the special human language widely used in the community of network communication via platforms such as online chat rooms/tools, bulletin board systems (BBS), email systems, blogs, etc. NIL is ubiquitous due to the rapid proliferation of Internet applications. In particular, chat text is a popular NIL genre which appears frequently in chat logs of online education (Heard-White, 2004) and customer relationship management (Gianforte, 2003) via chat rooms/tools. In wed-based chat rooms and BBS a large volume of NIL text is abused by solicitors of terrorism, pornography and crime (Finkelhor et al., 2000; McCullagh, 2004). A survey by the Global System for Mobile Communication (GSM) showed that Germans send 200 million messages a year (German News, 2004). All the facts disclose the rising importance in processing chat language text.

Chat language holds anomalous characteristics in forming non-alphabetical characters, words, and phrases. For example, "b4" is used to replace "before" in English NIL text and "94(jiu3 si4 in Chinese Pinyin)" to replace "就是(jiu4 shi4, exactly be)" in Chinese chat language text. Such characteristics pose problems to conventional natural language processing (NLP) tools in handling chat language text. For example, the chat term "细 8 细(xi4 ba1 xi4)", which is used to replace "是不是(be or be not)", is segmented to three common words, i.e. "细(slim)", " 8 (eight)" and "细", with ICTCLAS tool (Zhang et al., 2003). Other types of errors also occur in processing chat language phrases. To obtain better effectiveness, processing chat language text requires adjusted or new techniques to extract and normalize the chat terms before conventional NLP tools are deployed. Preliminary experiments in (Xia et al., 2005) reveal that knowledge based approach, i.e. pattern matching, exhibits poor adaptivity when processing unseen chat terms. Instead, corpus based machine learning approaches appear to be more robust in processing text. This results in our work in constructing a chat language corpus which is used specially in developing and evaluating techniques in extraction and normalization of chat terms. This is the first Chinese informal language corpus available for informal language processing research. The corpus is named as NIL corpus. In this corpus we tag two types of chat terms, i.e. ones holding anomalous morphological forms and ones holding standard morphological forms but expressing anomalous meanings compared to formal human language.

The first issue in constructing the NIL corpus is data source. Due to privacy concerns, obtaining large scale real chat text is difficult. Fortunately, we have located BBS chat text within "大嘴区(da4 zui3 qu1, free chat zone)" in YESKY BBS system (http://bbs.yesky.com/ bbs/) which exhibits remarkable chat characteristics and contains a vast amount of chat terms. We download BBS chat text posted from December 2004 to July 2005 in this zone. We finally collected 12,112 pieces of chat language text containing 92,314 words and 12,983 chat terms. Annotating the NIL corpus requires special knowledge on chat language. This results in two problems in corpus annotation. Firstly, as chat language is a rather new text genre, most of our annotators feel difficult to identify the chat terms and determine their counterparts in standard Chinese. This causes enormous annotation inconsistencies. Secondly, the annotators take a great deal time in determining the appropriate counterpart for each chat term. This renders huge manpower.

To minimize manpower and annotation inconsistency, we devise a two-stage incremental annotation approach to creating the NIL corpus on a GUI-based annotation platform. In the first stage, we sort the raw chat language text pieces according to timestamp. The first 1000 pieces are annotated under the specification of NILEML (see Section 2) on the annotation platform. In the second stage, ten incremental annotation iterations are conducted. In each iteration an automated annotation module is trained on all available annotated chat language text pieces and applied to identify and annotate chat terms in the next 1000 pieces of chat language text. No doubt that quality of the auto-

mated annotation module can be improved in each next iteration. Efficiency of corpus annotation can be improved because most seen chat terms can be annotated automatically and some unseen chat terms can be recognized by this module. Thus, manpower is mainly involved in 1) justifying the automated annotation, and 2) identifying and annotating the unrecognized chat terms.

The remaining sections of this paper are organized as follows. The NILEML annotation scheme is presented in Section 2. In Section 3 we describe the annotation components in this corpus annotation task. In Section 4 we describe the two-stage incremental annotation approach. We present experiments as well as discussions in Section 5. We describe related works in Section 6 and conclude this paper in Section 7.

2. Annotation Components

2.1. Annotation Scheme

In this paper an XML-based annotation scheme, i.e. NILEML, is devised to annotate the chat language text. In NILEML scheme, NILEML is defined to tag the NIL text documents and NILEX to tag chat terms. NILEX entails attributes of chat terms including text string, class, counterpart in standard language text, part of speech (POS) tag, segments if it is a phrase, POS tags for all segments, and its Chinese Pinyin. The attributes are defined in Figure 1.

attributes ::= nid string class normal pos [segments] posseg] [pinyin]
nid ::= n<integer> ; universal ID number
string ::= CDATA ;string of the NIL expression
class ::= CDATA
{class ::= 'A'|'F'|'H'|'T'|'O'} ; class of the NIL expression
; A = Abbreviation, F = Foreign expressions
; H = Homophony, T = Transliteration
; O = Other classes
normal := CDATA ; normal text for the NIL expression
pos := CDATA
{pos := 'NOUN'|'PRON'|'VERB' |'ADJ'|'ADV'|'NUMBER'
|'UNIT'|'PREP'|'CONJ'|'AUX'|'EXCL'}
; POS tag for the NIL expression
segments := CDATA ; segments for the NIL expression
posseg := CDATA
{posseg := 'pos-1|…|'pos-k'}
; POS tag list for the segments for the NIL expression
pinyin := CDATA ; Chinese pinyin ;for the NIL expression

Figure 1. Definition of NILEX attributes.

For checking the XML tagging syntax conformity, a document type definition (DTD) file is created to properly specify NILEML and NILEX tag set.

Two issues should be seriously considered in defining tag set for NILEML. The first issues is completeness of the tag set. NILEML is currently a task-orientated annotation scheme. Thus only attributes used in recognition task are configured in the annotation scheme. At the same time, we try to cover most commonly used linguistic attributes such as word segments and POS tag. Justification of value set for each attribute is the second issue we should address. Due to observation limit, we are only able to define values we've encountered. For example, we define value set for class attribute to be {'A', 'F', 'H', 'T', 'O'} based on observation of 13,068 NIL expressions. There is no doubt that new values will appear in the future annotation. The treatment is that we either cover them by 'O' or represent them by a new value. XML allows that an annotation scheme can be extended easily.

2.2. Computer Aided Annotation Platform

Defining attributes of chat terms must be conducted by experts who are familiar with chat language even though a XML-tagged text can be created using text editors. However, not all experts are familiar with XML tags. To help the human annotators, we develop a GUI-based computer aided annotation platform.

On the platform, human annotators can be concentrating on defining linguistic attributes for the chat terms while the NILEML tag set is automatically managed beneath the interface. For example, when the annotator select HOMOPHONY for the chat terms in the dropdown list for class attribute, the class attribute, namely "class='H'", is inserted in to the current NILX tag automatically.

2.3. Automated Annotation Module

The automated annotation module integrates a SVM classifier which is trained on the annotated chat terms and used to identify chat terms automatically. When a terms is recognized, a search action is executed to check whether the terms appears in the annotated NIL corpus. The whole NILEX tag is duplicated as the tag of the terms if it already exists. Otherwise, an empty NILEX tag will be created by the platform and a human annotator is required to specify its attributes.

Chat term recognition is a binary classification task in the annotation module. We use the SVM^{light} (Joachims, 1998) in our SVM implementation. Features considered for each chat term in SVM classification are listed below.

1) The occurrence of chat term when its
- *string* appears in any word bi-grams or tri-grams,
- *POS tag* appears in any POS tag bi-grams or tri-grams;
- *POS tags for segments* appear in any POS tag bi-grams or tri-grams;

2) The Boolean value that indicates whether a chat term
- is a *number* (Chinese or Arabic);
- contains merely *Latin capitals*;
- contains more than two *standard Chinese words*;
- contains *punctuations*;
- mixes *Chinese character and number*;
- mixes *Chinese character and Latin characters*;

An input chat language text is first segmented using ICTCLAS tool. We then use the SVM classifier to process all sequential combinations of the segments. For example, the chat language text "这个人 8 错(zhe4 ge4 ren2 ba1 cuo4, This people is not bad)" is first segmented to "这个|人| 8 |错". Ten sequential combinations are processed by the SVM classifiers, i.e. {"这个", "这个人", "这个人 8 ", "这个人 8 错", "人", "人 8 ", "人 8 错", " 8 ", " 8 错", "错"}. For this case, " 8 错" is identified as a chat term by the SVM classifier. To reduce computational complexity, we choose to combine up to 4 standard words in chat term recognition.

3. Two-Stage Annotation Approach

Basic idea of the two-stage incremental annotation approach is described as follows. In the first stage, we first sort the raw chat language text pieces according to timestamp. We annotate the first 1000 pieces of chat language text manually. In the second stage, ten incremental annotation iterations are conducted with the automated annotation module (see Section 2.3). In each iteration, we train the module on all available annotated chat language text and apply the module to annotate chat language text pieces in the next blocks. Workflow for the incremental annotation is shown in Figure 2.

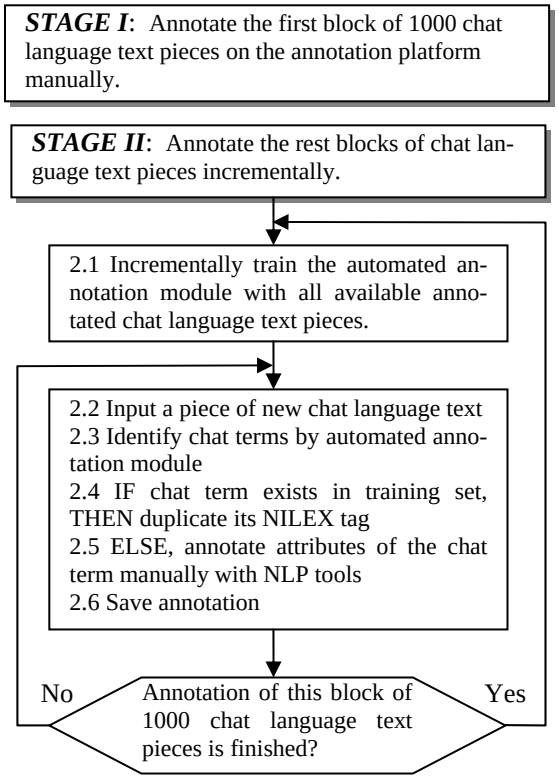

Figure 2. Workflow of the two-stage NIL corpus annotation.

3.1. Stage I: Manual Annotation

We first sort the raw chat language text pieces according to timestamp. The we split all text pieces into eleven blocks in which each block contains 1000 pieces. As the last block contains 112 pieces, we merge the last two blocks into one. We annotate the first block of 1000 pieces of raw chat language text under the specification of NILEML the annotation platform.

Chat terms are identified from chat language text manually. Attributes for chat term are specified by human annotators. To improve efficiency and quality, conventional NLP tools are integrated to produce some attributes automatically.

For example, the annotators are required to assign one of four classes (i.e., 'A', 'F', 'H' and 'T') to each chat term. The equivalent formal language text for each chat term is also defined by them manually. Word segments and POS tags can be produced by ICTCLAS tool automatically. A Chinese Pinyin transcription tool is developed with on CEDICT (Denisowski, 2005) to produce standard Chinese Pinyin for Chinese characters.

Coordination between annotators in this stage is plentiful because every chat term is annotated for the first time. The annotators have to negotiate with each other on whether a piece of text string is a chat term and how its attributes are specified.

3.2. Stage II: Incremental Annotation

We develop an automated annotation module based on SVM machine learning technique. Ten annotation iterations are conducted as follows. In each iteration, we train the module on all available annotated chat language text pieces and apply it to annotate chat language text in every next block. That is, at the very beginning of this stage, the module is trained on 1000 pieces of annotated chat language text and applied to annotate the second 1000 pieces. When the second 1000 pieces are successfully annotated, 2000 pieces of annotated chat language text are available for training in next iteration. We then train the module incrementally on the 2000 annotated pieces and apply the updated module to annotated the third 1000 pieces. The incremental annotation iteration is repeated until all chat language text pieces are annotated.

It is not likely that all chat terms in the to-be-annotated chat language text pieces can be identified correctly because some unseen chat terms are not recognized due to the limited training data. But the recognized ones are rather helpful. We devise the annotation method as follows. If a recognized chat term appears within the annotated NIL corpus, the previous NILEX tag is duplicated as the tag for the recognized chat term. Otherwise, an empty NILEX tag will be created for this chat term and the human annotator is prompted to specify its attributes on the platform. As such, human annotators' part in the annotation work is summarized to be, 1) justifying the automated annotation, 2) annotating the identified unseen chat terms, and 3) identifying and annotating the unrecognized chat terms manually.

We believe that manpower of corpus annotation can be saved because a large number of seen chat terms can be duplicated automatically by the annotation module.

3.3. Annotation Consistency

Consistency is a serious problem for each annotation task. It entails inter-annotator agreement (i.e., one sentence is annotated by two annotators equally) and intra-annotator consistency (i.e., annotations for same sentences are equally by one annotator). Consistency is usually maintained during the whole annotation process in which several annotators are possibly involved. Usefulness of a corpus relies highly on consistency in training or testing automatic methods. To guarantee a satisfactory annotation consistency, we define some guiding annotation principles as follow.

- The annotators are strictly required to negotiate with each other to produce an agreed annotation for each new chat term.
- The annotators are strictly required to duplicate previous annotation to the same chat term recognized.
- When suggested, revision should be agreed by all annotators.

- When revision is finally conducted, annotation for same chat terms must be revised at the same time.

The restrictions are helpful to avoid inconsistency between annotators during corpus annotation. High intra-consistency and inter-annotator agreement are thus obtained to assure the quality of the annotation.

4. Evaluations

To evaluate how much the automated annotation module improves the efficiency of NIL corpus annotation, several experiments are conducted to simulate the aforementioned incremental annotation process and reproduce the performance of the module over different versions of the training set and the test set.

4.1. Experiment Setup

Similar to the incremental annotation process, we use the time-based incremental training/test data split strategy according to the timestamp of the chat language text pieces. In the annotation-ready NIL corpus we currently have 12,112 annotated chat language text pieces sorted with timestamp from December 2004 to June 2005. Ten experiments are conducted. We start the experiments from training the module on the available 1000 pieces of annotated chat language text and testing it with the second block of 1000 raw pieces. We repeat the training and testing processes until all blocks of raw pieces are processed. Training/test data for all experiments are presented in Table 1.

Exp. No.	# of training chat terms	# of test chat terms	# of seen test chat terms	# of unseen test chat terms
1	996	997	414	583
2	1992	998	494	504
3	2989	1000	564	436
4	3988	997	583	414
5	4984	1001	648	353
6	5984	998	702	296
7	6981	995	713	282
8	7975	992	764	228
9	8966	998	791	207
10	9963	996	861	135
11	11956	1112	999	113

Table 1. Training/test data description.

Coverage curves for seen and unseen chat terms are presented Figure 3. We find percentage of unseen chat terms decreases from 58.5% in experiment 1 to 10.2% in experiment 11.

We use precision, recall and F-1 measure to present quality of chat term recognition. The precision is defined as the percentage of chat terms recognized correctly in all recognized chat terms. The recall is defined to be the percent-age of chat terms recognized correctly in those annotated by human annotators.

4.2. Results

We run the evaluation process on the eleven versions of training/test set. The overall experimental results are presented in Table 2.

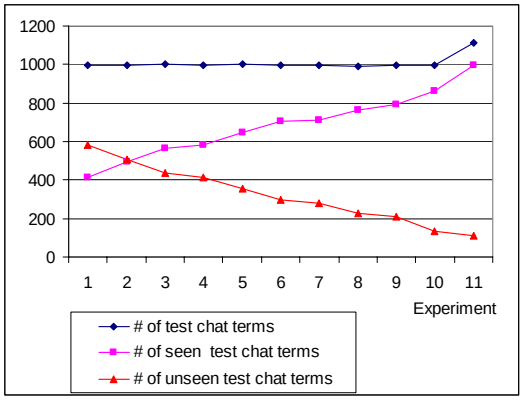

Figure 3. Coverage curves for seen and unseen chat terms in the 11 test sets.

Exp. No.	Precision	Recall	F-1
1	68.6	58.3	63.0
2	72.1	63.8	67.7
3	75.1	64.7	69.5
4	77.2	66.3	71.4
5	78.6	71.9	75.1
6	80.5	74.4	77.3
7	82.0	78.5	80.2
8	84.2	80.1	82.1
9	85.8	82.5	84.1
10	87.7	83.9	85.8
11	88.7	86.5	87.6

Table 2. Overall experimental results.

Exp. No.	Precision	Recall	F-1
1	88.9	77.5	82.8
2	87.9	82.4	85.0
3	89.4	78.1	83.4
4	89.7	81.0	85.1
5	88.4	87.3	87.9
6	88.0	87.9	88.0
7	89.9	92.5	91.2
8	89.1	91.8	90.4
9	90.4	90.7	90.6
10	90.8	89.1	89.9
11	91.0	89.6	90.3

Table 3. Experimental results for seen chat terms.

Exp. No.	Precision	Recall	F-1
1	54.2	45.2	49.3
2	56.8	47.6	51.8
3	56.7	48.0	51.9
4	59.7	48.0	53.2
5	60.6	48.7	54.0
6	62.5	49.2	55.0
7	62.1	50.6	55.7
8	67.5	51.2	58.2
9	68.1	56.4	61.7
10	67.4	56.2	61.3
11	69.2	62.9	65.9

Table 4. Experimental results for unseen chat terms.

It's natural that we split the test chat terms into the seen and the unseen. We refer the seen chat terms to the

ones that can be found in the training NIL corpus, and the unseen ones to those cannot be found in the training NIL corpus. We present results for the seen and unseen chat terms in Table 3 and Table 4 respectively.

4.3. Discussion I: Recognition Performance

We present the performance curves for precision, recall and F-1 measure in identifying all chat terms in Figure 4. It is observed that when the volume of training data increases, the overall performance is improved gradually. Note that evaluation was carried out within the same domain. This undoubtedly leads to high performance in the last several experiments.

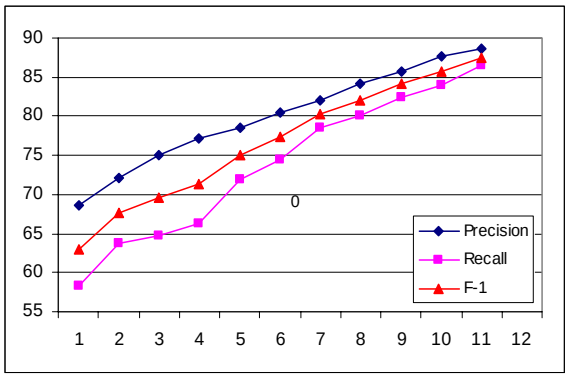

Figure 4. Quality curves for overall recognition.

We present quality curves in identifying seen and unseen chat terms in Figure 5 and Figure 6 respectively.

Curves in Figure 5 show that precision in identifying seen chat terms is relatively stable at around 90% in all experiments. This reveals that most seen chat terms can be correctly recognized. However, it is another story for performance of recall. In the first five experiment it climbs up from around 77% to 87%. This is reasonable because more training data normally leads to higher recall. Since in the last six experiments recall remains relatively stable, we may conclude that training data in experiment 5 is probably sufficient in identifying seen chat terms.

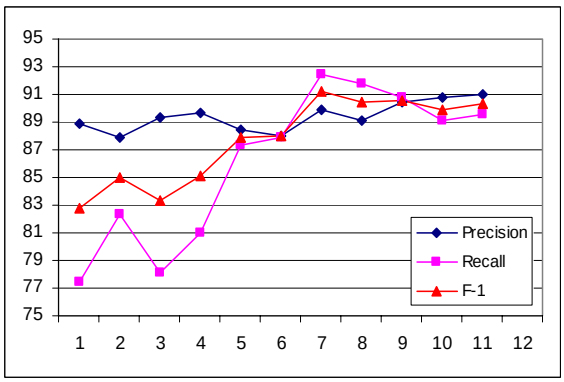

Figure 5. Quality curves for recognition of seen chat terms.

Identifying unseen chat terms is much more difficult than identifying seen ones with the SVM classifier. However, very encouragingly, the performance catches up when more training data is available according to Figure 6. Coverage curves for unseen chat terms in Figure 3 show that training data in experiment 11 is near to a sufficient volume in identifying unseen chat terms.

4.4. Discussion II: Annotation Efficiency

We find that the annotation efficiency can be improved in two manners with the two-stage incremental annotation approach. On the one hand, around 90% seen chat terms can be identified from chat language text correctly. Their NILEX tags can be duplicated without any changes. Human annotators' efforts can be reduced to justifying automated annotation and identifying unrecognized chat terms. A large volume of repetitive annotation work is therefore avoided.

On the other hand, the SVM classifier produces increasingly better quality in terms of correctly recognizing unseen chat terms. For example, around 70% unseen chat terms are identified correctly. The recognition output is helpful to alarm the human annotators, thus alleviate their work in picking out potential chat terms quickly. Efforts thus can be saved in carrying out this painstaking work.

We consider manual annotation of the first block of chat language text as experiment 0. Thus time used in annotating the twelve blocks of chat language text is presented in Table 5.

Exp. No.	Minutes on seen chat terms	Minutes on unseen chat terms	Total minutes
0	0	5031	5031
1	184	1580	1764
2	221	1431	1652
3	264	1236	1500
4	253	1236	1489
5	272	1070	1342
6	319	925	1244
7	332	876	1208
8	326	770	1096
9	348	705	1053
10	390	455	845
11	449	391	840

Table 5. Annotation time (minutes) used in twelve experiments.

Time used in manually annotating the first block of 1000 chat language text pieces in experiment 0 is 5031 minutes, namely around 5 minutes per chat term. According to Table 5, annotation time is reduced to 0.76 minute per chat term. It is thus proved that the efficiency is improved by 85.0% in annotating the last 1112 chat language text pieces.

4.5. Error Analysis

We summarize two types of typical recognition errors occurring in our experiments.

Err.1 Ambiguous chat terms

Chat text always contains common words with same characters as some chat terms. For example, when used in "答谢粉丝(da2 xie4 fen3 si1, thank the fans)", "粉丝(fen3 si1, vermicelli made from bean starch)" equals to 'fans'. But when used in "今天吃粉丝(jin1 tian1 chi1 fen3 si1, eat vermicelli today)", it is just a kind of food material. Such ambiguity also occurs frequently for the

numbers. 40 errors with this type happen in our experiment 11.

Err.2 Unseen chat terms

New chat terms come into birth very quickly. When the unseen chat terms provide no clue (e.g., satisfying any feature), the SVM classifier is incapable in those cases. 5 errors with this type happen in our experiment 11 including "盒饭(he2 fan4, takeaway food)" which represents 'fans of He Jie' (He Jie is a Chinese girl who got widely known recently in a TV show) .

5. Related Works

Corpus annotation is a prerequisite for many machine learning methods in chat language text processing but suffers from high cost and inter-annotator inconsistency. An interactive annotation approach is devised in (Thorsten and Oliver, 2000) in tagging and parsing the NEGRA corpus. Suggestions are produced automatically by a Cascaded Markov Models. The model is able to calculate reliability of the suggestions, and the annotator is prompted for confirmation or correction of the unreliable assignments. Such a semi-automatic process facilitates a very rapid and efficient annotation. However, tagging and parsing capability remains static during the corpus annotation process. We argue that the annotated text can be very helpful in improving tagging and parsing capability.

The feasibility of incremental linguistic annotation is examined in (Halteren, 1997). The article encourages reuse of annotated corpora already in existence and urge sufficient care should be taken with ambiguity, consistency and correctness in the incremental annotation. It is also argued that the feasibility of such an increase depends heavily on the way in which the incremental annotation is implemented. The two-stage incremental annotation approach is enlightened by the principle of incremental annotation. However, we discard the ambitious solution, i.e. fully automated incremental annotation. In our approach, we reduce the amount of human involvement as much as possible. Human efforts are expected to be concentrating on justifying ambiguity, consistency and correctness.

6. Conclusions

To minimize manpower and annotation inconsistency, we devise a two-stage incremental annotation approach to construct a chat language corpus. The first block of 1000 pieces of NIL text pieces regarding timestamp are annotated by human annotators in the first stage. In the second stage an automated annotation module is incrementally trained on all annotated chat language text available and applied to identify and to annotate chat terms in every next block of 1000 chat language text pieces. Two conclusions can be drawn from our experiments. One, with increasing volume of annotated NIL text pieces, quality of automated annotation of incoming chat language text pieces can be improved to around 88.7% in terms of precision. Two, the efficiency of corpus annotation is improved by 85.0% with the automated annotation module because more than 90% seen chat terms and more than 50% unseen chat terms can be annotated correctly with the annotation module.

7. Acknowledgements

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) and by Research Grants Council of Hong Kong under CERG project (PolyU 5181/03E).

8. References

Denisowski, P.. 2005. CEDICT: Chinese - English Dictionary. http://www.mandarintools.com/cedict.html.

Finkelhor, D., K. J. Mitchell and J. Wolak. 2000. Online Victimization: A Report on the Nation's Youth. Alexandria, Virginia: National Center for Missing & Exploited Children, page ix.

German News: Germans are world SMS champions, 8 April 2004, http://www.expatica.com/source/site_article.asp?subchannel_id=52&story_id=6469.

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.

Halteren, H. van. 1997. The Feasibility of Incremental Linguistic Annotation. http://www.cs.queensu.ca/achallc97/papers/p019.html.

Joachims, T.. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. ECML'98, pp. 137-142.

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

Heard-White, M., G. Saunders and A. 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.

Thorsten, B. and P. Oliver. 2000. Interactive Corpus Annotation. In Proc. of LREC 2000.

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

Zhang, Z., H. Yu, D. Xiong and Q. Liu. 2003. HMM-based Chinese Lexical Analyzer ICTCLAS. SIGHAN'03 within ACL'03, pp. 184-18.