Problematic Situation Analysis and Automatic Recognition for Chinese Online Conversational System

Yang Xiang[†], Yaoyun Zhang, Xiaoqiang Zhou[‡], Xiaolong Wang, Yang Qin
Key Laboratory of Network Oriented Intelligent Computation,
Harbin Institute of Technology Shenzhen Graduate School, China
[†]**windseedxy@gmail.com** [‡]**xiaoqiang.jeseph@gmail.com**

Abstract

Automatic problematic situation recognition (PSR) is important for an online conversational system to constantly improve its performance. A PSR module is responsible of automatically identifying users' un-satisfactions and then sending feedbacks to conversation managers. In this paper, we collect dialogues from a Chinese online chatbot, annotate the problematic situations and propose a framework to predict utterance-level problematic situations by integrating intent and sentiment factors. Different from previous work, the research field is set as open-domain in which very few domain specific textual features could be used and the method is easy to be adapted to other domains. Experimental results show that integrating both intent and sentiment factors gains the best performance.

1 Introduction

Automatic conversational systems are computer programs that interact with human users based on their knowledge bases. Developers of conversational systems devote plenty of efforts and time in collecting and verifying knowledge so as to maximize the information needs of potential users. However, problematic situations are inevitable due to several reasons (i.e. human verifiers would make mistakes or omissions, or quality of some answers couldn't be judged without certain contexts). So it is necessary to equip a conversational system with an automatic PSR module to keep its performance constantly improved. The program is responsible of monitoring whether the dialogue or some utterances are problematic during interactions and then providing feedbacks to the dialogue managers.

Problematic situations reflect that a human user is not satisfied with answers that a conversational system offers. From one perspective, some of these un-satisfactions can be captured through a human user's dialogue acts. For example, if a user repeats requesting the same question or frequently changes topics, it is likely that the system provides unsatisfactory answers (Chai et al., 2006). From another perspective, some explicit manners (i.e. sentiment-related expressions or dissatisfied feelings) that reflect the change of a user's mentality would also indicate a problematic situation occurs. Some previous systems use surveys to capture users' satisfactions: they let users to vote or evaluate whether the system has perfectly help them complete certain tasks (Hastie et al., 2002; Higashinaka et al., 2010) so as to collect users' satisficing scores. However, for a real-world conversational application, there are very few users who are willing to provide this kind of feedbacks.

The dialogue materials for this research come from a Chinese online chatting robot—BIT, which is developed for chatting and entertainment. It also integrates real-time data query functions about share price, weather report, post-code and telephone area code lookup. In addition to queries about real-time data, the corpus is totally open-domain and the number of topics that a dialogue could be related is unlimited. We annotated problematic situation labels in the utterance level (whether a question-answer pair is problematic/whether an answer is problematic) and took a deeper analysis towards different cases. Finally, we introduce the PSR framework. This framework is simple but efficient: we mapped the user intent and user sentiment categories to two groups of representative features and predicted problematic situations with supervised learners.

Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 43–51,
Wuhan, China, 20-21 October 2014

Our main contributions stem from the features, domains and language: Unlike most previous researchers who considered only user intent (Chai et al., 2006) or took offline satisfaction scores provided by users as user sentiment (Hastie et al., 2002; Higashinaka et al., 2010), our method integrates intent and sentiment in an online manner, which automatically identifies these two factors and gives the managers real-time feedbacks. The domain of the dialogue is open which is different from (Hastie et al., 2002; Chai et al., 2006). Another contribution is that this is the first work that solves this issue on the Chinese language, which has very different language specific features and resources from English.

We experimented on the corpus through 10-fold cross validation. In each individual fold, we compare our method with two baselines and with four popular classifiers. Results show that integrating both user intent and user sentiment factors gains the best performance with an average F_1 of 0.62 (by SVM).

Following, we first introduce related work o PSR from different perspectives. Introduction to the corpus are arranged next. The feature selections and the recognition framework are proposed in Section 4. Experiments, future work and conclusions constitute the rest.

2 Related Work

Previous researches in this literature differed in research grains, input features and research domains.

2.1 Dialog-level vs. Utterance-level

Most early work focused on the prediction of a complete dialogue. Hastie et al.(2002) predicted problematic dialogues from a series of DARPA Communicator dialogues according to user satisfaction rates, task completion predictors and some interaction based features. Walker et al. (2002) presented their prediction model on the basis of information the system collected early in the dialogue and in real time. Oulasvirta et al. (2006) reported relations between users' satisfaction rates among the goal-level, concept-level, task-level and command-level, and captured a number of qualified user features. Möller et al. (2008) evaluated performance of different models including linear regression models and classification trees on predicting dialog-level user satisfaction in three spoken dialogue datasets.

Although the predictions of progress towards dialogue completion might be used as a cue to the dialogue manager, the results couldn't reflect in which position a dialogue began to become problematic. Chai et al. (2006) proposed the definition of user intent and incorporate a few matching features to predict utterance-level problematic situations (whether an immediate answer is satisfactory). Engelbrecht et al. (2009) employ the Hidden Markov Model (HMM) to model the whole dialogue into a sequence where each node of the sequence corresponds to the quality of the utterance. Higashinaka et al. (2010a; 2010b) also use HMM to model the good/bad sequence and testing the effects of turn-wise and overall ratings. Similar spirit also exists in (Hara et al., 2010). Support Vector Machines (SVM) are used by Schmitt et al. (2011) for the quality prediction on the CMU's Let's Go Bus Information system (Raux et al., 2006) and ASR features are compared in their experiments.

2.2 Features

There are many factors that could affect the performance of judging whether a dialogue is problematic or not, i.e. time attributes like the total time of a dialogue and the time delays between utterances (Hastie et al., 2002; Walker et al., 2002; Möller et al., 2008), dialogue acts that may reflect user intents (Hastie et al., 2002) and users' satisfaction ratings toward the system's performance (Hastie et al., 2002). To avoid the side effects by Automatic Speech Recognition (ASR) and concentrate on the pure textual features in dialogues, several researchers only study the effect of dialogue acts and users' satisfaction ratings (Chai et al., 2006; Higashinaka et al., 2010). However, it has also proved that users' satisfaction ratings could not be always relied on since different groups of users may have different predictive powers (i.e. from novices to experts) (Möller et al., 2005).

2.3 Research Domains

Another main difference among previous researches is domain restriction. Specific domains or tasks simplify the PSR task and features are easy to be defined by employing domain experts. However, this restriction limits the ability of feature adaption from certain domains/tasks to others. In a way, domain-specific knowledge and user surveys are not easy to be adapted. As far as we know, most previous related work restricted their researches on specific domains such as travel plan making (Hastie et al., 2002), restrict-

ed scenarios (Chai et al., 2006), bus schedule information (Schmitt et al., 2011), music information (Hara et al., 2010), animal discussion and attentive listening (Higashinaka et al., 2010a; Higashinaka et al., 2010b).

3 Problematic Situation Analysis

This section will first introduce the characteristics of the corpus we construct and then provide definitions and examples for what we have learned from the dialogues.

3.1 Corpus Description

The corpus includes 479 dialogues with totally 3111 QA pairs. The dialogues are extracted from log files of the BIT robot from May to June, 2013. Each dialogue has a specific session ID, identifying that the dialogues are collected from different users or on different time. Chatting (> 2/3), stock real-time inquiries (<1/6) and weather report inquiries (<1/7) account for the largest proportion. The dialogues are almost original which contains a number of curse words (although we have removed some too dirty words), facial expressions (by expressing moods through several punctuations such as ":)"), boring statements (i.e. I am boring uh) as well as duplicate questions, indicating the irregular and informal characteristics of the online chatting contexts. The language of the corpus is Chinese, with very few English utterances (<1/100). The length of dialogues ranges from 1 to 64 QA pairs[1].

To collect users' un-satisfactions, the developers place good/bad comment buttons for each answer but it seems to be useless (seldom users would click the buttons). Through observation, we found that users' behaviors, including the type of both *inquiries* and *responses*, provide important cues to determine whether a user is satisfied with an answer or not. To show this relation, two examples are listed in Table 1.

We asked two annotators to label whether each answer is problematic or not. They gave their judgment according to whether they considered the answer provided by the chatting system was reasonable or not, but ignored whether it was not precise (i.e. a factual answer that is out of date). When labeling the current pair, they were asked to refer to the above interactions so as to take the context restrains into consideration. By using Cohen's kappa coefficient, the inter agreement of the two annotators is $\hat{P}_o \approx 0.933$, $\hat{P}_e \approx 0.602$, and $\kappa \approx 0.833$ which is relatively high. Most conflicts occur when the two annotators have different under standings towards a question or one of them couldn't understand what a question really means. The conflict labels are resolved by a third annotator. Finally, 832 out of 3111 pairs (26.7%) are labeled as problematic, indicating that the chatting system still has a lot of room for improvement.

Type.	Question by User	Answer by BIT	Explanation
Sentiment-related	你会…? (Can you do…?)	这也不会,那也不会(I can't do anything.) -- problematic	The second question which is a curse sentence implies that the user is not satisfied with the robot's previous answer.
	你真是个废物 (You are really a good-for-nothing.)	我是比特 (I am BIT.)	
Intent-related	你儿子呢? (Where's your son?)	你要闹哪样啊? (What're you going to do?) --Problematic	Adjacent utterances. Repeat questions reflect the unsatisfactory of the user towards the last answer. Probably due to irrelevant answers. On the left, the first answer is irrelavant.
	你儿子呢? (Where's your son?)	天若赐我辉煌, 我必比天猖狂 (If the God favours me, I would be crazier than the God.[Chinese network catchwords]) -- Problematic	

Table 1. Examples of problematic situations in BIT.

3.2 Corpus Deeper Analysis

According to observation, the style of sentences raised by users could be roughly divided into two groups: *questions* and *statements*(corresponding to *inquiries* and *responses* in the previous section). Questions are sentences that send inquiries to the system, indicating that users have some information needs. Contrarily, statements are sentences that reflect no information needs, but could express complains, exclamations or some other affections. We've also observed that a specific group of features is much related to questions while another group is more likely to co-occur with statements.

[1] In this paper, we will use the notion "utterance" and "QA pair" exchangeablely.

	Category	Utterance	Explanation
Int.	switch	1. 中国(China)	The current question belongs to a different topic from the last one. The beginning of a new **dialogue** (other than greeting) is classified to switch.
	retry	2. 中华人民共和国 (People's Republic of China)	The current question has the same idea as the last one but may be expressed in a different style.
	continue	3. 中国首都 (The capital of China)	The current question belongs to the same topic as the last one. The example is a detailed question about the topic "China".
	clarify	4. 中国首都在哪里? (Which city is the capital of China?)	Negotiate with the system to refine or coarsen the last question for a clearer intent.
Sen.	greeting	早上好(Morning) / 亲爱的! (Honey!)	Usually a beginning or ending of a dialog. Intimate speeches are also categorized into greeting.
	criticize/ response	你好聪明!(You are so clever!) / 你说对了(You are right)	Criticism or response towards the last answer. Positive or negative criticisms frequently occur in the corpus, indicating users' (un)satisfactions.
	exclaim/ statement	好烦啊!(It's so boring!) / 我喜欢**. (I love someone.)	Exclaims or statements that the user delivers which are not aiming at the chatbot.
	curse	Dirty words.	Explicit curse words that are inevitable in chatting dialogues. They sometimes show unsatisfactory, but sometimes occur dues to that the user has been ridiculed by the robot.
	order	讲个笑话!(Tell me a joke!)	Order the system to provide information or do something.
	other	。。。/ !!!	Utterances other than the above such as punctuations or symbols that might show speechless(。。。), exclaiming / warning(!!!) or some facial expressions.

Table 2: Examples and definitions for user intent (Int.) and user sentiment (Sen.).

Based on this intuition, we define two concepts as:

User Intent – the action of a user when raising a question, indicating that the user is executing an inquiry to the system.

User Sentiment – the sentiment or affection that a user expresses through his/her utterances, including negative and non-negative.

The definition of user intent follows (Chai et al., 2006). It mainly contains four lower-level types: *switch, continue, retry,* and *clarify*. Switch means to start a new topic or a new dialog. Continue, retry and clarify are restricted in the same topic, with different dialogue acts. User sentiment is associated with the following cases: *greeting*, *criticize/response*, *exclaim/statement*, *curse*, *order* and *other*. *Other* contains punctuations, facial expressions and special symbols that are frequently used in Chinese daily chatting. Examples with explanations for user intent and sentiment are listed in Table 2.

The annotations towards the lower-level categories have more conflicts (with an average κ about 0.5) than the problematic labels. The disagreements are solved after declaring some issues: 1) if intent and sentiment characteristics both occur, label according to the type of the sentence (question correlates with intent and statement with sentiment) 2) *Criticize*s are towards the system's last response while *curse*s are not.

Problematic situations that originate from the following types are more direct and easier to understand: a) repeat the last question (retry, 4.95%-45.45%); b) change the topic (switch, 32.27%-32.17% with 6.97% at the beginning); c) try to clarify what the user intended to ask (clarify, 1.29%-50%) d) negative criticisms towards the last answer (criticize, 13.79%-15.85%); e) negative words toward the robot (curse, 6.59%-11.7%). The percentages 4.95%-45.45% stand for that *retry* accounts for 4.95% in all, and among all the *retry* cases, 45.45% are problematic. We also have the polarity (negative or non-negative) of each user provided utterance annotated and find that nearly all the negative occur in statements. The rest problematic situations mostly come from the *other* type (8.61%-48.13% with 36.19% facial expressions that the system is not able to recognize), *continue* (8.01%-28.4%), *exclaim* (10.83%-19.58%) and *order* (7.23%-19.55%).

We also notice that in several cases, although users hadn't received satisfactory answers, they didn't mean to negotiate with the system any more, indicating that many users are not patient enough to provide cues. These cases bring about difficulties for the prediction. Another special case we notice is from the disagreements of annotators, that is, sentiment and intent characteristics could co-occur (i.e. repeated *curse*s). This

inspires us to synthesize both user intent and user sentiment attributes for an utterance.

4 Recognition Framework for Problematic Situations

Based on a simple dependency analysis for a dialogue, we first map user intent and user sentiment into related feature groups, and then use the features to predict problematic situations.

4.1 Utterance Dependency Analysis

A dialogue could be modeled using a directed graph constituted by the question sequence ***Q*** and the answer sequence ***A***. In the graph, a node stands for an utterance (question/answer), and edges are drawn from each Q_{i-1} to Q_i, Q_i to A_i, A_{i-1} to A_i and A_{i-1} to Q_i. The edges stand for dependencies or constrains between utterances (Figure 1).

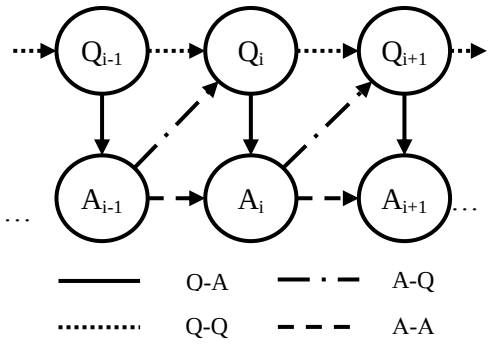

Figure 1. Dependencies or constrains in dialogues

In this work, the edges from A_{i-1} to Q_i and from Q_{i-1} to Q_i are the main dependency types we research. A_{i-1} to Q_i shows the last answer affects the current question in a dialog, always reflected by user sentiment. Constrains between questions are more related to user intent, i.e. the current question would have a high similarity with the last one if one attempts to retry an inquiry. The following example typically shows the two types of constrains:

Q_{i-1}: Who did you go with yesterday?

A_{i-1}: My advantage is that I am handsome.

Q_i: Who did you go with yesterday?

A_i: If the God favours me, I would be crazier than the God.

Q_{i+1}: You are an idiot.

In the example, the *retry* case from Q_{i-1} to Q_i implies that A_{i-1} should not be a good answer. The negative *curse* Q_{i+1} indicates that A_i may be problematic.

4.2 Mapping to problematic situations

To avoid cascade errors brought about by lower-level classifications, we weaken the category constrains by mapping the taxonomy to related features. The four types for user intent could be distinguished by features considering about similarity between sentences, which descends from *retry*, *clarify* to *continue* and *switch*. For the six types in user sentiment, we define word features, word polarity features and pattern features to make the types distinguished.

In our proposed framework, the automatic PSR problem is simplified into a one level binary classification task in which utterances are modelled with general features, user intent specific features and sentiment specific features. General features are textual and non-textual features that have nothing to do with user intent or user sentiment, including: *whether the answer is from the system's default response list to underdeveloped knowledge*, *whether the question is a real-time inquiry*, *the number of utterances before and follow* (especially to distinguish the beginning or ending of a dialog), *the similarity between the question and its corresponding answer*.

User intent specific features are those extracted from the perspective of user intent, mainly related to the similarity between two adjacent questions. User sentiment specific features are those extracted from the perspective of user sentiment, which focus on whether a user-raised utterance contains any sentiment information.

4.3 Intent Specific Feature Selection

Specifically, we tag whether the current question is *retry* because *retry* always corresponds to a very high similarity which is easy to be identified and many of them are related to problematic situations. We also use the similarity between two questions to distinguish the other types of intents. Typical features are listed in Table 3(NE stands for Name Entity).

The semantic similarity measure between questions (labeled by * in Table 3) is based on a Chinese semantic web, HowNet (Dong and Dong, 2006). The defined semantic similarity in HowNet is a normalized real value ([0,1]) of the shortest path connecting two words in the HowNet Concept Relation Net. Suppose two questions *P* and *Q* (word sequence size *m* and *n*, respectively), the semantic similarity between them is defined as:

$$ssim(P,Q) = \frac{1}{2}(\frac{\sum_{i=1}^{m} P_i}{m} + \frac{\sum_{j=1}^{n} Q_j}{n})$$

where P_i and Q_j are denoted as:

$$P_i = \max(ssim(P_i, Q_1), ssim(P_i, Q_2), ..., ssim(P_i, Q_n))$$

$$Q_j = \max(ssim(P_1, Q_j), ssim(P_2, Q_j), ..., ssim(P_m, Q_j))$$

ssim(P_i,Q_m) denotes the semantic similarity of the *i*th word in *Question P* and the *m*th word in *Question Q*. If two words are the same, the similarity is set to *1*.

The final similarity is defined as:

$$sim(P, Q) = \lambda_1 nsim(P, Q) + \lambda_2 ssim(P, Q)$$

nsim(P,Q) is the normalized real value of the number of words the two questions share. λ_1 and λ_2 are the weighted parameters (set to be *0.5*,*0.5* in our experiment).

Feature	Description
Exact match (Boolean)	After removing punctuations and stop words.
No. of NEs	By analyzing results of LTP.
NE similarity	The match No. and contents for NEs.
Ques. Similarity	Weighted similarity based on lexicon and semantics*.
Ques. similarity without NEs	Weighted similarity based on lexicon and semantics*.
Target word	The target word in a question.
Dependency similarity	Dependency pattern similarity.

Table 3. User intent specific features.

The target words, name entities and dependency trees are identified or generated by LTP (LTP, Liu et al., 2011). Target words are defined as the direct objects that the root verb governs in a dependency parse tree in questioning sentences. The dependency similarity is computed by counting the number of common dependency relations (normalized to [0,1]).

4.4 Sentiment Specific Feature Selection

User sentiment is a good reflection of a user's current mood. The difficulty lie on that *curse* sentences and negative *criticisms* are not easy to be distinguished, especially for the Chinese language where many sentences have no subjects at all. A solution is that considering both the similar key words between the last answer and the current statement, and whether a second person pronoun (i.e. you/BIT) exists.

This work models the possible relations from sentiments to problematic situations by defining a series of sentiment related features. We employ dictionary-based method (Zhao et al., 2010) to judge the polarity of words in a sentence. Typical features are shown in Table 4.

Feature	Example
Key words	弱智(stupid), 次(weak)
Question word/ question mark	为什么(why), 是什么(what), 是谁(who), ?
Target word	天气(weather), 人名(person name)
Ending word	好吗(is it ok?), 吗(modal)
Sent. pattern	你好/真/太傻(you're quite/very/too stupid)
Part-of-Speech	Adjectives, nouns
Polarity	Polarity of a word
Person pronoun	你(you), 比特(the name of the robot)
Dependency	Subject-verb-object (SBV and VOB by LTP)

Table 4. User sentiment specific features.

Cursing sentences or negative criticisms are usually expressed in certain patterns which could be captured through regular expressions after removing adverbs and modals. Adjective and noun words are good indicators for sentiment which could be looked up in sentiment dictionaries. We employ two general Chinese sentiment dictionaries (NTUSD[2] and HowNet) to determine the polarity of a word (including both nouns and adjectives for the consideration of both *You're a fool* and *You're foolish.*). In addition, we tag the sentence as *negative* if it only contains negative words (key words) after removing useless components. Real-time inquiries are special cases that we should filter out through key words matching.

There are also something special that we should consider. Suppose there are three continuous pairs: A->B->C: If the question in B contains negative criticism information but A is a real-time inquiry, we couldn't directly judge A is problematic. A typical example is that the answer is closely related but is not precise (i.e. out of date). Inquiry includes questions about weather,

[2] http://nlg18.csie.ntu.edu.tw:8080/lwku/pub1.html

stock, post code, telephone and identity code in this system.

In addition to un-satisfactions for not achieving the desired answer, *curse/criticism* sentences could also grow out from some other cases: (1) the user has been ridiculed by the system thereby becomes irritated; (2) the user just wants to express his/her feelings to the system through repeated statements. These cases are not directly related to problematic situations, which, however, haven't been well recognized yet, hindering the improvements of the learners.

4.5 The Recognition Framework

We expected that the lower-level category information could be well modeled through features and classifiers. General features, user intent specific features and user sentiment specific features are extracted for each QA pair. Intuitively, the feature groups for user intent and user sentiment have relatively different emphasis and the hybrid features should naturally increase the system's recall.

Suppose the sequences *are* $\boldsymbol{Q}$ and $\boldsymbol{A}$, in which Q_i is to be determined (see Figure 1). The automatic PSR model is described as the follows:

a) Pre-processing: tokenization, POS tagging, parsing, removing stop words, and filtering system specified inquires (weather, stock, post code, telephone and identity code);
b) Extract sentiment specific features for Q_i based on Q_i;
c) Extract intent specific features for Q_i based on Q_{i-1} and Q_i;
d) Tag whether Q_i is *retry* or not, tag whether Q_i is *negative* or not;
e) Determine problematic of Q_i according to sentiment (*retry* or not) and intent labels (*negative* or not), specific features (Table 3 and 4) and general features (§4.2), as well as the labels for Q_{i-1} (*retry*, *negative*, and *problematic*);
f) Post-processing: For the last QA pair in a dialog, if a same pair exists before and is labeled as problematic, Q_i is labeled problematic.

The reason why we also take the labels of Q_{i-1} into account is based on the fact that the labels of Q_{i-1} may help determine the current label. For example, if the last intent indicates a *retry* and the current question indicates a *switch* (a much lower similarity with the last one), it is very likely that the user has tried at least twice but hasn't received a satisfactory answer. In this case, the previous *retry* could also increase the probability of *switch*, which is helpful for the final determination.

Post processing mainly deals with the last utterance in a dialogue which doesn't have any followings.

5 Experiments and Analysis

To prove the effectiveness of our model, we compare it with two baselines on four classical classifiers through 10-fold cross-validation.

The baselines include the model with general features (GF) and intent specified features (ISF), the model with GF and sentiment specified features (SSF). We name our hybrid model that with hybrid features as GF+ISF+SSF. We report the detailed performance gains of the GF+ISF+SSF model compared with the two baselines with intense experiments on the corpus. General features (GF) only contains little useful information towards our task and has very poor performance, therefore we didn't set it as a baseline. We test the model with SVM, Naïve Bayes, Decision Tree and CRF so as to find out an efficient and stable learner for the task.

GF+SSF			
	Prec.	**Rec.**	**F_1**
SVM	92.97	44.44	60.05
J48	85.03	22.94	35.85
NB	95.37	22.53	36.37
CRF	89.20	40.06	55.01
GF+ISF			
	Prec.	**Rec.**	**F_1**
SVM	93.77	43.80	59.57
J48	88.76	21.72	34.67
NB	**96.39**	23.24	36.42
CRF	88.74	44.89	59.46
GF+ISF+SSF			
	Prec.	**Rec.**	**F_1**
SVM	85.73	49.38	**62.19**
J48	79.15	24.89	37.75
NB	85.97	29.09	43.35
CRF	91.08	**45.02**	60.16

Table 5. Average performance by cross-validation.

10-fold cross validations are performed on the dataset. To specify, the corpus should be divided in the unit of dialogues rather than utterances for the sake of integrating sequential features (i.e. the previous labels). LibSVM (Chang and Lin, 2011), Naïve Bayes and Decision Tree (J48) were provided by the Weka toolkit (Hall, et al.,

fold	Prec.	Rec.	F_1	Percent.	Best	Learner	im-in	im-sen
1	89.21	49.01	63.27	26.69	intent	CRF	-0.03	+6.1
2	84.57	50.51	63.25	30.24	hybrid	SVM	+3.16	+1.74
3	92.41	42.07	57.82	29.51	hybrid	CRF	+0.12	+6.38
4	85.47	47.52	61.08	28.85	hybrid	SVM	+3.16	+3.16
5	84.82	54.92	66.67	28.41	hybrid	SVM	+3.61	+3.61
6	95.42	44.17	60.39	25.94	sentiment	SVM	-0.07	-0.07
7	78.45	55.69	65.14	26.90	hybrid	SVM	+3.12	+3.72
8	85.33	52.46	64.97	32.50	hybrid	SVM	+2.93	+3.42
9	83.03	51.31	63.43	26.90	hybrid	SVM	+5.47	+4.0
10	94.29	39.87	56.05	28.55	sentiment	SVM	+1.3	-0.2

Table 6. Detailed results in 10-fold cross validation.
"im-in" and "im-sen" stand for the improvements of the hybrid model than intent and sentiment specific models. "Percent." stands for the proportion% of problematic utterances in this fold of data.

2009). CRF is provided by CRF++[3], a C++ implementation. Metrics of *precision*, *recall* and F_1 are used for evaluation.

We list results for the average performance of cross-validation in Table 5. From the data we notice, all the four learning models perform well in precision but a little poor in recall (no matter for which model). And the case of Naïve Bayes is especially obvious. According to analysis towards the output, the performance of high precision and low recall mainly due to the following reasons: Firstly, we select features empirically which may generate strong rules: if some condition is satisfied, some conclusion is drawn. Secondly, there are still a number of situations that we couldn't resolve by training our models. For example, not all *retry* result in problematic situations, and sometimes the users' intents are hard to understand. Finally, there are many negative sentences that are not related to problematic situations which could confuse the learners.

We also notice that SVM and CRF have much better results than J48 and Naïve Bayes, implying the effectiveness of the two classifiers. The hybrid model outperforms the two baselines mainly by recall, reflecting the reasonability of considering both user intent and sentiment.

More evidence for the robustness of the hybrid features and the learners can be recognized through a detailed report of the cross validation (Table 6). From the table we observe two important things: one is that SVM performs much more stable than other classifiers, and CRF is not so good as what we have expected, considering there are sequential features; the other is that the hybrid model outperforms other baselines in most cases, and it also has comparative results in other cases (fold 1, 6, and 10).

What we have also noticed is that although Naïve Bayes doesn't achieve a better score in F_1, it always performs well in precision (Table 5). Its characteristics of running fast, easy implemented and with high precision enable the developers to integrate the automatic recognizer in the system and send back precise predictions in real time.

6 Future Work

We left two problems for future work. Firstly, although we have defined lower-level categories for user sentiment and user intent, we failed to well identify each of them. More representative features (maybe word embedding or something else) should be extracted to clearly identify their boundaries. Secondly, there is much noise in the original corpus which may affect the model performance. An automatic sieve should be developed to deal with the noisy information.

7 Conclusion

This paper analyses different problematic situations under the chatting context for the Chinese language. Other than previous work, we propose the problematic situation recognition model from two perspectives—user sentiment and user intent, and test the proposed model on a totally open-domain corpus. Experiments verify that integrating both the two factors gains the best predicting result. More representative features and more efficient approaches will be developed for further improvement.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 612-72383 and 61173075). And the foundations of

[3] http://crfpp.googlecode.com/svn/trunk/doc/index.html

Shenzhen(JC201005260118A, ZDSY20120613-125401420, JCYJ20120613151940045, and JC201005260175A).

Reference

Joyce Y. Chai, Chen Zhang, and Tylor Baldwin. 2006. Towards Conversational QA: Automatic Identification of Problematic Situations and User Intent. In Proceedings of COLING/ACL.

C. C. Chang and C. J. Lin. 2011. LIBSVM : A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, 2(27)1-27.

Zhendong Dong, and Qiang Dong. 2006. HowNet and the Computation of Meaning. River Edge, NJ: World Scientific, 25-76.

Klaus-Peter Engelbrech, et al. 2009. Modeling User Satisfaction with Hidden Markov Model. In Proceedings of the SIGDIAL 2009 Conference. Association for Computational Linguistics.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. 2010. Estimation Method of User Satisfaction Using N-gram-based Dialogue History Model for Spoken Dialogue System. LREC.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. 2009. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Helen Wright Hastie, Rashmi Prasad, Marilyn-Walker. 2002. What's the Trouble: Automatically Identifying Problematic Dialogues in

DARPA Communicator Dialogue Systems. In Proceedings of ACL.

Ryuichiro Higashinaka, et al. 2010. Issues in Predicting User Satisfaction Transitions in Dialogues: Individual Differences, Evaluation Criteria, and Prediction Models. Spoken Dialogue Systems for Ambient Environments. 48-60.

Ryuichiro Higashinaka, et al. 2010. Modeling User Satisfaction Transitions in Dialogues from Overall Ratings. In Proceedings of the SIGDIAL 2010 Conference.

Ting Liu, Wanxiang Che, Zhenghua Li. 2011. Language Technology Platform. Journal of Chinese Information Processing. 25(6): 53-62.

Sebastian Möller, et al. 2005. Quality of Telephone-based Spoken Dialogue Systems.

Sebastian Möller, Klaus-Peter Engelbrecht, and Robert Schleicher. 2008. Predicting the Quality and Usability of Spoken Dialogue Services. Speech Communication 50.8: 730-744.

A.Oulasvirta, S.Möller, S. Engelbrecht, et al. 2006. The Relationship of User Errors to Perceived Usability of a Spoken Dialogue System. In Proceedings of the 2nd ISCA/DEGA Tutorial and Research Workshop on Perceptual Quality of Systems, Berlin.

A. Raux, D. Bohus, B. Langner, A. W. Black, and M. Eskenazi. 2006. Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience. In Proceedings of the International Conference on Speech and Language Processing.

Alexander Schmitt, Benjamin Schatz, and Wolfgang Minker. 2011. Modeling and Predicting Quality in Spoken Human Computer Interaction. In Proceedings of the SIGDIAL 2011 Conference.

Marilyn A. Walker, et al. 2002. Automatically Training a Problematic Dialogue Predictor for a Spoken Dialogue System. Journal of Artificial Intelligence Research, Vol.16(1): 293-319.

Yanyan Zhao, Bing Qin, and Ting Liu. 2010. Sentiment Analysis. Journal of Software, 21(8): 1834-1848.DARPA Communicator Dialogue Systems. In Proceedings of ACL.