PREDICTING USER EVALUATIONS OF SPOKEN DIALOG SYSTEMS USING SEMI-SUPERVISED LEARNING

Baichuan Li[1†], *Zhaojun Yang*[2], *Yi Zhu*[1], *Helen Meng*[2], *Gina Levow*[3‡], *Irwin King*[1]

[1]Department of Computer Science and Engineering
[2]Department of System Engineering and Engineering Management
[1,2]The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[3]Department of Linguistics, University of Washington, Seattle, WA 98195 USA

ABSTRACT

User evaluations of dialogs from a spoken dialog system (SDS) can be directly used to gauge the system's performance. However, it is costly to obtain manual evaluations of a large corpus of dialogs. Semi-supervised learning (SSL) provides a possible solution. This process learns from a small amount of manually labeled data, together with a large amount of unlabeled data, and can later be used to perform automatic labeling. We conduct comparative experiments among SSL approaches, classical regression and supervised learning in evaluation of dialogs from CMU's Let's Go Bus Information System. Two typical SSL methods, namely co-training and semi-supervised support vector machine (S3VM), are found to outperform the other approaches in automatically predicting user evaluations of unseen dialogs in the case of low training rate.

***Index Terms*—** Spoken Dialog System, Semi-Supervised Learning, Evaluation

1. INTRODUCTION

With the advancement of speech and language technologies, there is an increasing number of SDSs in many information domains. An SDS is an intelligent system which can communicate with users and answer their questions through dialog interactions. Usually SDSs operate in closed information domains. An example is the CMU Let's Go! Bus Information System[1]. It works in the domain of bus information and it provides schedules and route information for the city of Pittsburgh's Port Authority Transit (PAT) buses. Figure 1 shows the text automatically transcribed from one dialog, from which we know that the user wanted to leave from Swissvale and travel to Dynamo Way, and would like to know the departure time of the next 61A bus. The SDS recognized the user's idea and responds with relevant information.

System: Welcome to the CMU Let's Go bus information system.
System: To get help at any time, just say Help or press zero.
System: What can I do for you?
User: NEXT 61A
System: The 61A. Did I get that right?
User: YES
System: Alright
System: Where are you leaving from?
User: .FIFTH AT
System: For example, you can say, NORTH BRADDOCK, FORBES AND MURRAY, or DOWNTOWN.
User: SWISSVALE
System: Leaving from SWISSVALE. Is this correct?
User: YES
System: Okay
System: What is your destination?
User: DYNAMO WAY
System: Going to DYNAMO WAY.
System: Hold on. I'll look that up.
System: The next 61A leaves NOBLE AVENUE AT WASHINGTON at 7:35 p.m. and arrives at DYNAMO WAY AT BEECH at 7:55 p.m.

Fig. 1. An example dialog in the Let's Go! system

An important research question is how one may evaluate the quality of an SDS, as well as compare different SDS. In general, the system's quality is mainly determined by the quality of dialogs it generates. In the PARADISE framework [12], the quality of a dialog is measured by task success and dialog costs. Additionally, users may also give their feedback about the SDS, which is important for gauging the quality of the dialogs. We assume that a dialog engenders good user experiences (and thus has good quality) if the SDS can correctly understand the user's intention and provide relevant and useful information.

In this regard, a straightforward technique to estimate the quality of a dialog is to extract a number of interaction parameters and integrate these parameters using a regression model to get a holistic rating. To be specific, we can first obtain user evaluations for some dialogs (this process is called labeling) and use these dialogs as training data. Then we can calcu-

[†] The corresponding author.

[‡] This work is done while Dr. Gina Levow was a visiting scholar at The Chinese University of Hong Kong.

[1] http://www.speech.cs.cmu.edu/letsgo/.

late the parameters of the regression model. The model can further be used to estimate user evaluations for new unseen dialogs.

However, when the amount of training data is small, the performance of regression models may not be good enough. Furthermore, there are two problems in labeling. The first is that it is costly to manually label large dialog corpora. The second problem is the variability in user evaluation. When one dialog is evaluated by multiple users, different people may give different evaluations for the same dialog.

The first problem may be addressed through using semi-supervised learning (SSL) [16]. The advantage of SSL is that it can make use of a small amount of labeled data with a large amount of unlabeled data to predict the labels of unlabeled data. In order to investigate the effectiveness of SSL, we apply two typical SSL approaches (namely, S3VM and co-training) in predicting user evaluations for unlabeled dialogs.

The second problem may be addressed through utilizing crowdsourcing [15], where we use *Amazon Mechanical Turk*[2] to collect users' experiences, followed by the design of a framework to deal with the variability of user evaluations.

This paper presents our latest effort in applying SSL to predict user evaluations of SDS. We also test SSL algorithms on a real dialog corpus and draw performance comparisons with classical regression models and supervised learning approaches. To our knowledge, we are the first one to explore SSL in predicting user evaluations of SDS.

The paper is organized as follows. In Section 2, we provide a brief overview of recent related work on both spoken dialog system evaluation and semi-supervised learning. Section 3 details how the semi-supervised learning methods are applied to predict user evaluations. In Section 4 and Section 5, we describe the experimental setup and analysis of results. Conclusions are given in Section 6.

2. RELATED WORK

SDS evaluation has been developed for many years and a number of frameworks have been proposed. Walker et al. [12] proposed a general evaluation framework called PARADISE which measures the quality of SDS from two aspects: task success and dialog costs. In this framework, a good SDS should maximize completion rates while minimizing dialog costs. Thereafter, Hassel and Hagen [5] improved the PARADISE framework by overcoming the limitations of requiring unimodality and also a clear task description in the form of an attribute-value-matrix (AVM). In addition, Cózar et al. [7] and Griol et al. [4] presented approaches to evaluate SDSs through user simulation techniques. Möller and Ward [9] further proposed a tripartite framework to evaluate SDS: "One part models the behavior of user and system during the interaction, the second one the perception and judgment processes taking place inside the user, and the third part models what matters to system designers and service providers."

Similar to our work, Evanini et al. [3] used a decision tree to predict caller experience, and Engelbrech et al. [2] use Hidden Markov Models to predict the user judgements. However, the main differences from our work are: (1) we utilize crowdsourcing rather than experts [3] or true users [2] to get more annotated dialogs; (2) we leverage SSL to predict user evaluation, which is more suitable when the amount of labeled data is small.

At the same time, researchers are discovering more and more interaction parameters [6] about the quality of SDSs. Möller gave an overview of these interaction parameters in [8].

SSL is a class of machine learning techniques that make use of a small amount of labeled and large number of unlabeled data for training. Different from supervised learning which only trains classifiers with labeled data, SSL can leverage the information of unlabeled data and perform better than supervised learning in most cases, especially when the size of labeled data is very small.

S3VM and co-training are two typical SSL methods. S3VM was first proposed by Vapnik [11] and co-training was originally proposed by Blum and Mitchell [1].

SSL has been applied in many areas. Xu et al. [14] exploited a semi-supervised text categorization framework by active search. Tang and his colleagues [10] applied S3VM in visual tracking. Recently, Wan [13] proposed a co-training approach for cross-lingual sentiment classification.

3. LEARNING TO PREDICT USER EVALUATION

This section begins with stating the problem of predicting user evaluation using SSL, with labeled and unlabeled data. Then we describe how the EM, S3VM, graph-based SSL and co-training are applied to predict user evaluations.

3.1. Problem statement

Recall that our objective is to predict user evaluation for a dialog. For convenience, we treat it as a binary classification problem, i.e., users' evaluations are either good or bad. Formally, let the feature vector $\mathbf{x} \in \mathcal{R}^D$ denote the parameters of a dialog and $y \in \{1, 0\}$ represent the users' evaluations (1 means good, 0 otherwise). In this paper, we use $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ to denote the set of labeled dialogs (labeled data), in which l is the size of the set, **x** is feature vector and y is the corresponding user evaluation on this dialog. $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$ are used to represent the set of unlabeled dialogs (unlabeled data) whose size is u. Our goal is to predict the labels for the unlabeled data $\{\hat{y}_i\}_{i=l+1}^{l+u}$ and make the predicted values as accurate as possible.

[2] http://www.mturk.com.

3.2. S3VM

Assume that the user's evaluation is linearly related to the features of the dialog and let

$$\hat{y} = f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \tag{1}$$

where $w \in \mathcal{R}^D$ is a weight vector and $b \in \mathcal{R}$ is an offset parameter. The S3VM [11] objective is defined as:

$$\min_{\mathbf{w},b}\sum_{i=1}^{l}\max(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b), 0) + \lambda_1\|\mathbf{w}\|^2 +$$

$$\lambda_2\sum_{i=l+1}^{l+u}\max(1 - |\mathbf{w}^T\mathbf{x}_i + b|, 0), \tag{2}$$

subject to

$$\frac{1}{u}\sum_{i=l+1}^{l+u}\mathbf{w}^T\mathbf{x}_i + b = \frac{1}{l}\sum_{i=1}^{l}y_i, \tag{3}$$

in which the first term $\sum_{i=1}^{l}\max(\cdot)$ is the loss function for labeled data, the second term $\|\mathbf{w}\|^2$ is regulation term and the third one $\sum_{i=l+1}^{l+u}\max(\cdot)$ is the loss function for unlabeled data. λ_1 and λ_2 are two weighting coefficients.

We use the tool UniverSVM[3] to implement the objective function above and give the predictions of user evaluations on unlabeled dialogs.

3.3. Co-training

The automatically extracted features provide multiple views for dialogs. For instance, according to the ITU-T P-series Recommendations [6], there are five classes of parameters (synonymous with features):

- Dialogue- and communication-related parameters;
- Meta-communication-related parameters;
- Cooperativity-related parameters;
- Task-related parameters;
- Speech-input-related parameters.

Empirically these classes of features are mutually independent, which suits the assumption of view independence in co-training quite well.

The main idea of co-training is simple. Given two views of data, we train two different classifiers with few labeled instances based on each view (the specific learning algorithms used can be the same). Then each classifier is applied to the unlabeled instances and the most confident candidates are moved from the unlabeled data set to the labeled data set. Thus, labeled instances are augmented and new classifiers can be trained with expended labeled data. The above process will be repeated until the unlabeled data is used up or when some stopping criterion is satisfied (such as iteration times is finished). Table 1 presents the procedure of co-training.

Table 1. Co-training for predicting user evaluations on dialogs

Algorithm 1 Co-training for predicting user evaluations on dialogs

```
Input: labeled data L={$\mathbf{x}_i, y_i$}$_{i=1}^{l}$, $\mathbf{x}_i = \{\mathbf{x}_i^1, \mathbf{x}_i^2\}$, unlabeled
    data U={$\mathbf{x}_i$}$_{i=l+1}^{l+u}$, learning rate $p$, $n$, end condition $con$.
Output: Predicted labels for unlabeled data $\{\hat{y}_i\}_{i=l+1}^{l+u}$.
while con is not satisfied do
    Learn the first classifier $\mathbf{C_1}$ that considers only the $\mathbf{x}^1$ portion of $\mathbf{x}$
    Learn the second classifier $\mathbf{C_2}$ that considers only the $\mathbf{x}^2$ portion of $\mathbf{x}$
    Add $\mathbf{C_1}'s$ most confident $p$ positive predictions and $n$ negative predictions on U to L
    Add $\mathbf{C_2}'s$ most confident $p$ positive predictions and $n$ negative predictions on U to L
    remove these $2p + 2n$ instances from U
end while
```

4. EXPERIMENTAL SETUP

This section describes our experimental setup. First we present our dataset including dialogs and manual labels for each dialog. Second, we introduce the features we adopted. Then, we describe a few compared regression methods and supervised learning methods. At last we provide the evaluation metrics, which are used to evaluate each method's performance.

4.1. Data collection

We use the dialog data obtained from CMU's Let's Go Speech Dialog Database[4]. We published more than 10,000 dialogs's texts on *Amazon Mechanical Turk (MTurk)* and designed a questionnaire which contains 5 questions to get user evaluations on those dialogs. In order to guarantee the quality of human evaluations, we require the Worker's approval rate should be higher or equal to 98%. In addition, we manually designed several rules for the approval of ratings. For instance, we reject the rating for which the working time is less than 10 seconds. For more information, please refer to our work in [15]. Through validation of the user-evaluated data, we selected 4,907 dialogs together with their user evaluations based on the question "Do you think the system is successful in providing the information that the user wanted?" The question allows user responses on a five-point scale: entirely unsuccessful (1), mostly unsuccessful (2), half successful/unsuccessful (3), mostly successful (4) and entirely suc-

[3] http://www.kyb.mpg.de/bs/people/fabee/universvm.html/.

[4] http://www.speech.cs.cmu.edu/letsgo/letsgodata.html/.

cessful (5). The reason for choosing this question as user evaluation is that the inter-rater agreement for this question is the highest [15].

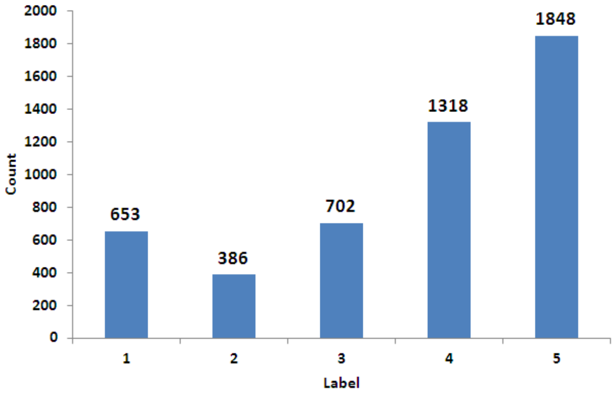

Fig. 2. Distributions of users evaluations based on the question "Do you think the system is successful in providing the information that the user wanted?"

Table 2. Selected dialog features based on ITU-T P-series Recommendations Supplement 24

Feature	Description	Type
# system turns	overall number of system turns	Integer
# user turns	overall number of user turns	Integer
WPUT	average number of words per user turn	Float
# system questions	overall number of questions from the system	Integer
# user questions	overall number of questions from the user	Integer
aveUserSpeakRate	average speak rate of users	Float
DTMF%	proportion of dual tone multiple frequency	Float
# barge-in	overall number of user barge-in attempts	Integer
aveRecogConf	average recognition confidence	Float
# help request	overall number of user help requests	Integer

Figure 2 gives the distribution of user evaluations on those 4,907 dialogs. We find that most users have high opinions of this system — nearly 65% of dialogs receive the score of 4 or above. On the basis of the above observation, we divide these dialogs into two categories: good ($y = 1$ for those dialogs whose scores are 4 or above) and bad ($y = 0$ for those dialogs whose scores are lower than 4). After splitting, there are 3,166 "good" dialogs and 1,741 "bad" ones.

4.2. Feature selection

According to the ITU-T P-series Recommendations Supplement 24 [6], we extract 10 quality-related features (parameters) automatically from these dialogs. Table 2 gives the description of these features. The first 6 features are dialogue- and communication-related interaction parameters while the others are meta-communication-related interaction parameters. For co-training algorithm, these two kinds of parameters form two views of each dialog naturally. All parameter values are converted into the range of [0,1] in data preprocessing.

4.3. Comparisons

We want to get the answer to the research question: "Can SSL give better performance in predicting user evaluation when the labeled dialogs is limited?" in our experiments. Thus, we compare the SSL methods with the following popular and frequently-used regression models and supervised learning algorithms:

- **Linear Regression**: the user's evaluation of a dialog is modeled as a linear function of the parameters, i.e., $y = \beta^T\mathbf{x} + b$, where β are coefficients of linear regression models and b is the intercept.

- **Logistic Regression**: we use a linear model to model the log-odds of the probability p that a dialog is good, i.e., $\frac{p}{1-p} = \beta^T\mathbf{x}$, where β are coefficients of linear models.

- **k-Nearest Neighbor (KNN)**: we classify unlabeled dialogs based on the closest training examples and assign labels with the class most common amongst the k nearest neighbors. In our experiments we set $k = 9$ after 10 fold cross validation.

- **SVM**: we predict the user's evaluation using an SVM classifier trained with the training data.

All the above methods only use labeled dialogs to build classifiers. Specifically, for regression models, we set a threshold $\tau = 0.5$ to transform the final regression values into two classes, i.e., if regression result for a unlabeled dialog is above 0.5, we classify it as good, otherwise we classify it as bad.

4.4. Evaluation metrics

We adopt Precision, Recall, Accuracy and F-score as metrics to evaluate the performance of SSL algorithms as well as regression and supervised learning methods in predicting user evaluations:

$$Precision = \frac{TP}{TP + FP}, \quad (4)$$

$$Recall = \frac{TP}{TP + FN}, \quad (5)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}, \quad (6)$$

$$F\text{-}score = \frac{2 \cdot TP}{2 \cdot TP+FP+FN}, \quad (7)$$

where TP = true positive, FP = false positive, TN = true negative, and FN = false negative.

5. EXPERIMENT RESULTS

We present the performance of each algorithm when training rate (i.e., the proportion of dialogs used as training data) is only 0.3%, which just provides enough data for regression models. Then we explore the impact of training rate on these algorithms. For each experiment, we split the dialogs into training data (labeled) and testing data (unlabeled) randomly according to the training rate and the ratio between good dialogs and bad ones (3166:1741), and run each algorithm with these data. We repeat the above process 10 times for a particular training rate and then calculate the average value as final results.

Table 3 reports the Precision, Recall, Accuracy and F-score for S3VM and co-training as well as compared methods when only 0.3% of dialogs (i.e., 9 good dialogs and 5 bad ones) are labeled. From this table we could see that in SSL methods, co-training (with logistic regression as inner classifier) performs better than S3VM, and both SSL methods outperform the other algorithms significantly. For instance, the Accuracy and F-score of S3VM are 7.96% and 7.66% relatively higher than those of SVM respectively. For supervised learning approaches, SVM outperforms KNN and both of them perform better than regression models. The two regression models have the similar low performance. Among all approaches, co-training performs best both in Accuracy and F-score. In particular, it achieves the Accuracy of 75.86% at such a low training rate, which is 18.81%, 19.18%, 15.22% and 12.40% relatively higher than the Accuracy of linear regression, logistic regression, KNN and SVM. The above results demonstrate the superiority of SSL in leveraging unlabeled data in the learning process.

Figures 3 and 4 show the Accuracy and F-score of SSL and compared approaches across different training rates. We note that SSL methods always perform better than others when the training rate is lower than 3% (i.e., there are less than 150 labeled dialogs among all 4,907 dialogs). When the training rate increases, two regression models obtain better performance. In our experiments, when the training rate is above 5%, the Accuracies of regression models are higher than SSL methods. The F-scores of each method is basically correlated with the Accuracy, except KNN. We find that KNN's Accuracy is the lowest in most cases among all methods, however, its F-score is higher than regression models and SVM when training rate is lower than 0.6%. The reason is than KNN owns high TP but very low TN, which makes it have low Accuracy but higher F-score. From the above two figures we can get the conclusion that SSL methods such as co-training and S3VM are most suitable for situations where there are very few labeled dialogs. When we have sufficiently many labeled dialogs, classical regression models can make sufficiently good prediction.

When we get the prediction of user evaluation for each dialog, we can utilize these evaluations together to give an overall evaluation of the system. One possible way is assigning each dialog with a normalized weight according to its typicalness in the system (for instance, one kind of dialog may be the commonest, thus it should be given the highest weight.) and then calculate the weighted sum of user evaluations of these dialogs as the holistic evaluation of the system.

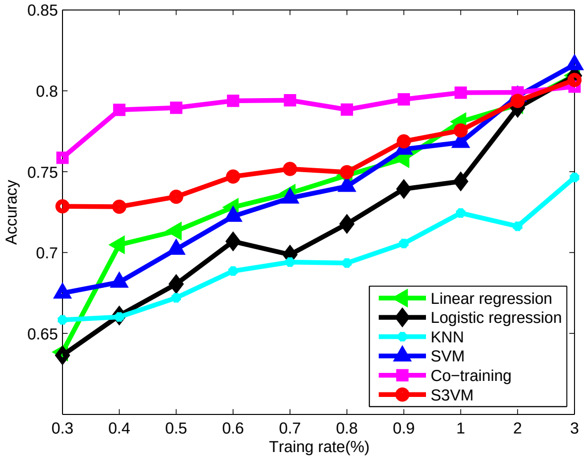

Fig. 3. Accuracy versus training rates across different methods

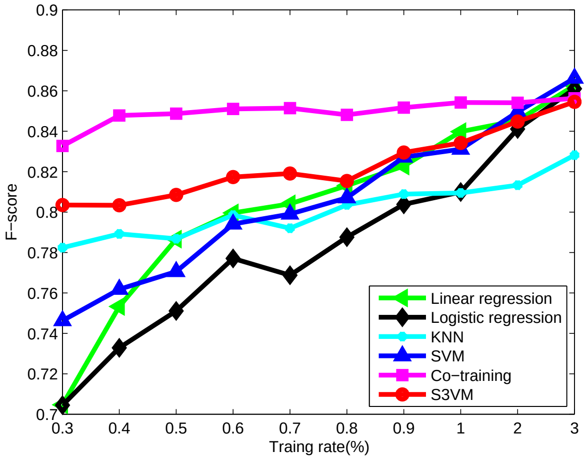

Fig. 4. F-score versus training rates across different methods

6. CONCLUSIONS

In this paper we apply SSL algorithms, namely S3VM and co-training, to predict user evaluations of SDS. We conduct

Table 3. Comparative performances of S3VM, co-training and other methods when training rate is 0.3%. The values in bold are the highest among all methods (each row). The baseline method predicts all unlabeled dialogs as "good" ones.

	Baseline	**Linear Regression**	**Logistic Regression**	**KNN**	**SVM**	**Co-training**	**S3VM**
Precision	0.645	0.6826	0.6855	0.6562	0.7443	**0.7594**	0.7541
Recall	**1.000**	0.7353	0.7322	0.9904	0.7581	0.9250	0.8600
Accuracy	0.645	0.6385	0.6365	0.6584	0.6749	**0.7586**	0.7286
F-score	0.784	0.7047	0.7045	0.7892	0.7463	**0.8327**	0.8035

experiments using a dataset with nearly 5,000 manual labeled dialogs and compare the performance of SSL with that of classical regression models and supervised learning methods. Our experiment results show that SSL methods perform much better than regression models and supervised learning approaches in predicting user evaluations when the number of labeled dialogs are very limited. For regression models or supervised learning methods, it is costly to manually label abundant training dialogs when a great number of dialogs are collected (e.g., for the purpose of SDS evaluation). However, SSL provides a good solution for dialog (also SDS) quality evaluation under such circumstances while only a small number of labeled dialogs are needed.

Future work includes further investigation in applying this evaluation methodology to conduct SDS evaluation among different SDSs. We will also attempt to transform the dichotomous labels (0 and 1) to multiple level ones and explore whether SSL still perform well.

7. ACKNOWLEDGMENTS

The work is partially supported by grant from the MSRA, FY09-RES-OPP-103 (Reference No. 6902682). It is also supported by NSFC/RGC Joint Research Scheme (Project No. N_CUHK 414/09).

8. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of COLT*, 1998.

[2] K.-P. Engelbrech, F. Gödde, F. Hartard, H. Ketabdar, and S. Möller. Modeling user satisfaction with hidden markov model. In *Proc. of SIGdial*, 2009.

[3] K. Evanini, P. Hunter, J. Liscombe, D. Suendermann, K. Dayanidhi, and R. Pieraccini. Caller experience: A method for evaluating dialog systems and its automatic prediction. In *Proc. of SLT*, 2008.

[4] D. Griol, L. F. Hurtado, E. Sanchis, and E. Segarra. Acquiring and evaluating a dialog corpus through a dialog simulation technique. In *Proc. of SIGdial*, 2007.

[5] L. Hassel and E. Hagen. Evaluation of a dialogue system in an automotive environment. In *Proc. of SIGdial*, 2005.

[6] International Telecommunication Union. ITU-T P-series Recommendations Supplement 24: Parameters describing the interaction with spoken dialogue systems, 2005.

[7] R. López-Cózar, A. De la Torre, J. C. Segura, and A. J. Rubio. Assessment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40(3):387–407, 2003.

[8] S. Möller, P. Smeele, H. Boland, and J. Krebber. Evaluating spoken dialogue systems according to de-facto standards: A case study. *Computer Speech and Language*, 21(1):26 – 53, 2007.

[9] S. Möller and N. Ward. A framework for model-based evaluation of spoken dialog systems. In *Proc. of SIGdial*, 2008.

[10] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *Proc. of ICCV*, 2007.

[11] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[12] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella. Paradise: a framework for evaluating spoken dialogue agents. In *Proc. of ACL/EACL*, 1997.

[13] X. Wan. Co-training for cross-lingual sentiment classification. In *Proc. of ACL-IJCNLP*, 2009.

[14] Z. Xu, R. Jin, K. Huang, M. R. Lyu, and I. King. Semi-supervised text categorization by active search. In *Proc. of CIKM*, 2008.

[15] Z. Yang, Y. Zhu, B. Li, I. King, G. Levow, and H. Meng. Collection of user judgements on spoken dialog system with crowdsourcing. In *Proc. of SLT*, 2010.

[16] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.