User Intent Prediction in Information-seeking Conversations

Chen Qu
University of Massachusetts Amherst
chenqu@cs.umass.edu

Liu Yang
University of Massachusetts Amherst
lyang@cs.umass.edu

W. Bruce Croft
University of Massachusetts Amherst
croft@cs.umass.edu

Yongfeng Zhang
Rutgers University
yongfeng.zhang@rutgers.edu

Johanne R. Trippas
RMIT University
johanne.trippas@rmit.edu.au

Minghui Qiu
Alibaba Group
minghui.qmh@alibaba-inc.com

ABSTRACT

Conversational assistants are being progressively adopted by the general population. However, they are not capable of handling complicated information-seeking tasks that involve multiple turns of information exchange. Due to the limited communication bandwidth in conversational search, it is important for conversational assistants to accurately detect and predict user intent in information-seeking conversations. In this paper, we investigate two aspects of user intent prediction in an information-seeking setting. First, we extract features based on the content, structural, and sentiment characteristics of a given utterance, and use classic machine learning methods to perform user intent prediction. We then conduct an in-depth feature importance analysis to identify key features in this prediction task. We find that structural features contribute most to the prediction performance. Given this finding, we construct neural classifiers to incorporate context information and achieve better performance without feature engineering. Our findings can provide insights into the important factors and effective methods of user intent prediction in information-seeking conversations.

KEYWORDS

User Intent Prediction; Information-seeking Conversations; Conversational Search; Multi-turn Question Answering

ACM Reference Format:
Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. 2019. User Intent Prediction in Information-seeking Conversations. In *2019 Conference on Human Information Interaction and Retrieval (CHIIR '19), March 10–14, 2019, Glasgow, Scotland Uk.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3295750.3298924

1 INTRODUCTION

Many companies have launched conversational assistants (CA) such as Amazon Echo, Google Home, Microsoft Cortana, etc. These devices allow users to issue voice commands to the CA for goal oriented tasks or to conduct simple question answering (QA) tasks, such as adding calendar events or asking for news. This trend has led many researchers in the information retrieval (IR) and natural language processing (NLP) community to pay more attention to conversational search. For examples, SIGIR'18, ICTIR'17, and EMNLP'18 all have workshops[1,2,3] on conversational search.

However, most CAs are not yet capable of handling multi-turn information-seeking conversations, where users have multiple rounds of information exchange with CAs to retrieve or specify answers. One reason is the difficulty to model the conversation about the information need both before and after an answer has been given. Thus an important step in modeling conversational interactions is to accurately detect and predict user intent in this interactive information-seeking process. More specifically, CAs should be capable of improving previous answers if they can correctly process critical information provided by the users, such as relevance judgments (feedback) and clarifications of the information need. Thus, CAs need to elicit more information proactively when they are not confident, before providing an answer [24].

The Learn-IR workshop[4] at WSDM'18 highlighted the significant research need for user intent analysis and prediction in an interactive information-seeking process. To address this research demand, we conducted experiments on user intent prediction using the MSDialog [19] data. This data collection consists of multi-turn information-seeking dialogs in the technical support domain. The dialogs are typically initiated by an information seeker who asks technical issues about Microsoft products, such as "How do I downgrade from Windows 10 to Windows 7?". This kind of question is non-factoid and often requires multiple rounds of conversational interactions. The answers are provided by Microsoft staff and other experienced product users such as "Microsoft Most Valuable Professionals" (MVPs). These human agents (information providers) also explicitly ask for feedback on their provided answers and thus keep the users engaged. The MSDialog data is annotated with a set of 12 *user intent types* [2, 19]. There are different definitions of "user intent" in our field. In this paper, user intent refers to a taxonomy of utterances in information-seeking conversations (Table 1). Each utterance of the MSDialog was annotated with the user intent types using Amazon Mechanical Turk (MTurk)[5]. MSDialog provides a high-quality resource to show how information-seeking conversations are structured between humans. In addition to MSDialog, we also used a portion of Ubuntu Dialog Corpus (UDC) [14] which was annotated with the same user intent types in [19] to further validate our findings.

CHIIR '19, March 10–14, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-6025-8/19/03…$15.00
https://doi.org/10.1145/3295750.3298924

[1] https://sites.google.com/view/cair-ws/cair-2018
[2] http://sigir.org/ictir2017/sessions/search-oriented-conversational-ai-scai/
[3] https://scai.info/
[4] https://task-ir.github.io/wsdm2018-learnIR-workshop/
[5] https://www.mturk.com/

The purposes of user intent prediction are threefold. First, it is necessary for CAs to accurately identify user intent in information-seeking conversations. Only in this way can CAs process the information accordingly and use it to provide answers and adjust previous answers. Similar to customer service over phones, routing user questions to different sub-modules in a conversational retrieval system is only possible if the user intent is correctly identified. Second, the CAs need to learn and imitate the behavior of human agents. By identifying user intent in information-seeking conversations, we expect the CA to learn the use of different intent and when to issue requests for more information or details spontaneously. Finally, user intent prediction models can be used to automatically annotate more dialog utterances for data analysis and other tasks such as conversational answer finding.

Previous work typically focused on dialog act classification for open-domain conversations [9, 13, 22]. In human-computer chitchat, the goal of the CA is to generate responses that are as realistic as possible with the primary purpose of entertaining. In contrast to chatting, users initiate information-seeking conversations for specific information needs. Human behaviors in chatting and information-seeking conversations can be very different due to the fundamentally distinct purposes. In addition, the Dialog State Tracking Challenges (DSTC)[6] focus on goal oriented conversations. These tasks are typically tackled with slot filling [27, 31]. In information-seeking conversations, slot filling is not suitable because of the diversity of information needs. User intent analysis and prediction are needed for an information-seeking setting.

We conduct experiments using two different approaches to predict user intent in information-seeking conversations. Firstly, we extract rich features to capture the content, structural, and sentiment characteristics of utterances and learn models with traditional machine learning (ML) methods. Secondly, we use the implicit representation learning in neural architectures to predict user intent without feature engineering. We then incorporate context information into neural models for enhanced performance.

Our contributions can be summarized as follows. (1) We extract rich features including feature groups related to content, structures, and sentiment to predict user intent in information-seeking conversations. We perform an in-depth feature importance analysis on both group and individual level to identify the key factors in this task. (2) We design several variations of neural classifiers to predict user intent without explicit feature engineering. We show that neural models can achieve comparable performance compared to feature engineering based methods. Moreover, neural models achieve statistically significant improvements over traditional methods after incorporating context information. (3) Our experiments show that the trained model achieves good generalization performance on another open benchmark information-seeking conversation dataset (UDC). The code of the implemented user intent prediction models will be released to the research community.[7]

The rest of our paper is organized as follows. In Section 2, we present related work regarding utterance type classification, conversational search, and multi-turn question answering. In Section 3, we formulate the research question of user intent prediction in information-seeking conversations. We also describe the data creation and annotating process. In Section 4, we extract rich features and learn traditional ML models for user intent prediction. We also perform feature importance analysis in this section. In Section 5, we introduce various enhanced neural classifiers for user intent prediction. Section 6 presents the conclusion and future work.

[6] https://www.microsoft.com/en-us/research/event/dialog-state-tracking-challenge/

[7] https://github.com/prdwb/UserIntentPrediction

2 RELATED WORK

Our work is closely related to utterance classification, conversational search, and multi-turn question answering.

Utterance Classification. Utterance classification is well studied in the NLP and IR domain. Many different classification techniques such as statistical approaches [22], SVM, or Hidden Markov Models [23] have been used for different applications including human-human chatting [22], student's utterance [17], or forum post classifications [2]. However, recent advances in deep learning allow us to use neural architectures for utterance classification both on the word [10] and character level [32]. These new deep learning techniques have been applied in medical dialog systems [4]. In this paper, we focus on user intent prediction in information-seeking conversations. This specific utterance classification task presents unique challenges because of the complexity and diversity of human information-seeking conversations.

Conversational Search. Searching via conversational interactions with IR systems is an increasingly popular research topic in both industry and academia [3, 21, 33]. Oddy [16] introduced man-machine IR through dialogs without explicitly formulating queries. Belkin et al. [1] modeled the human-computer interaction in information-seeking as dialogs. Moreover, information-seeking via conversations is especially important in exploratory search [15, 25], where users are unfamiliar with the domain they are searching and would rely on effective interactions with retrieval systems. More recently, Radlinski and Craswell [20] identified key properties in conversational IR systems. Trippas et al. [24] conducted lab-based observational studies on conversational search and identified that it is more interactive than traditional search and new information-seeking models are needed. In our work, we focus on an essential study in conversational search, which is to predict user intent in this information-seeking setting. Our findings can help build conversational search systems that can provide enhanced searching experience using predicted user intent.

Multi-turn Question Answering. Early research on multi-turn question answering dates back to AutoTutor [6], which can simulate human tutors to assist college students to learn computer science. Recent work has focused on single turn QA on factoid questions [30] and other open-domain questions (e.g. WikiQA [29]). The Ubuntu Dialog Corpus [14] and MSDialog [19] provide large scale multi-turn QA dialogs in the technical support domain. Wu et al. [26] and Yang et al. [28] used these datasets to perform conversation response ranking for non-factoid questions. Our work focuses on a specific research need for multi-turn QA in the information-seeking setting. The performance of multi-turn QA could be improved if the user intent is correctly identified.

3 TASK DEFINITION AND DATASET

3.1 Task Definition

The research problem of user intent prediction in information-seeking conversations is defined as follows. The input of the system is an information-seeking dialog dataset $\mathcal{D} = \{(\mathcal{U}_i, \mathcal{Y}_i)\}_{i=1}^{N}$ and a set of user intent labels $\mathcal{L} = \{l_1, l_2, \ldots, l_c\}$. A dialog $\mathcal{U}_i = \{u_i^1, u_i^2, \ldots, u_i^k\}$ contains multiple turns of utterances. u_i^k is the utterance at the k-th turn of the i-th dialog. $\mathcal{Y}_i$ consists of annotated user intent labels $\{\mathbf{y}_i^1, \mathbf{y}_i^2, \ldots, \mathbf{y}_i^k\}$, where $\mathbf{y}_i^k = \{y_i^{k(1)}, y_i^{k(2)}, \ldots, y_i^{k(c)}\}$. Here $y_i^{k(m)}, \ldots, y_i^{k(n)} = 1$ denotes that the utterance u_i^k in dialog $\mathcal{U}_i$ is labeled with user intent $\{l_m, \ldots, l_n\}$. Given an utterance u_i^k and other utterances in dialog $\mathcal{U}_i$, the goal is to predict the user intent $\mathcal{Y}_i$ of this utterance. The challenge of this task lies in the complexity and diversity of human information-seeking conversations, where one utterance often expresses multiple user intent [24].

3.2 Dataset

We use the MSDialog dataset that consists of technical support dialogs for Microsoft products. The data was crawled from the well-moderated Microsoft Community forum[8] which contains high-quality technical support dialogs between users and agents. The agents include Microsoft staff, community moderators, MVPs, or other experienced product users. Very rich structured data were collected with the dialogs, including the question popularity, answer vote, affiliation of information providers, etc. We choose this dataset because of its information-seeking nature and the well annotated user intent. Although MSDialog has limitations such as a narrow subject domain and forum-style language, no other openly available dialog datasets with the same detailed annotation exist. We believe this should be a first step to predict user intent in an information-seeking setting.

The dataset contains two sets, a complete set that consists of all the crawled dialogs and a labeled subset that contains only dialogs with user intent annotation. A taxonomy of 12 labels presented in Table 1 were developed in Qu et al. [19] based on work by Bhatia et al. [2] to characterize the user intent in information-seeking conversations. We also present the user intent distribution in Table 1. The complete set consists of 35,000 multi-turn QA dialogs in the technical support domain. Over 2,000 dialogs were selected for user intent annotation on MTurk with the following criteria: (1) With 3 to 10 turns. (2) With 2 to 4 participants. (3) With at least one correct answer selected by the community. (4) Falls into one of the categories of following major Microsoft products: Windows, Office, Bing, and Skype. The inter-rater agreement score was used to ensure the annotation quality. One utterance can be labeled with more than one user intent. A comparison of statistics between the complete set and the labeled subset is presented in Table 2.

In order to test the generalization performance of our findings, we use a small portion of UDC that is annotated with the same user intent types. This dataset also consists of multi-turn information-seeking conversations in a technical support domain between an information seeker and provider. However, UDC is generated from internet relay chat (IRC) and contains a significant amount of typos, Internet language, and abbreviations. In addition, UDC contains shorter utterances and more turns per dialog. This part of experiment is presented in Section 5.3.

[8] https://answers.microsoft.com

Table 1: User intent taxonomy and distribution in MSDialog

Code	Label	Description	%
OQ	Original Question	The first question from the user to initiate the dialog.	13
RQ	Repeat Question	Other users repeat a previous question.	3
CQ	Clarifying Question	User or agent asks for clarifications.	4
FD	Further Details	User or agent provides more details.	14
FQ	Follow Up Question	User asks for follow up questions about relevant issues.	5
IR	Information Request	Agent asks for information from users.	6
PA	Potential Answer	A potential answer or solution provided by agents.	22
PF	Positive Feedback	User provides positive feedback for working solutions.	6
NF	Negative Feedback	User provides negative feedback for useless solutions.	4
GG	Greetings/Gratitude	Greetings or expressing gratitude.	22
JK	Junk	No useful information in the utterance.	1
O	Others	Utterances cannot be categorized using other classes.	1

Table 2: Statistics of MSDialog (complete & labeled subset)

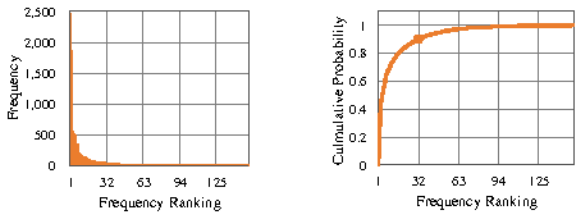
Items	Complete set	Labeled subset
# Dialogs	35,000	2,199
# Utterances	300,000	10,020
# Words (in total)	24,000,000	653,000
Avg. # Participants	3.18	2.79
Avg. # Turns Per Dialog	8.94	4.56
Avg. # Words Per Utterance	75.91	65.16

3.3 Data Preprocessing

The purpose of this classification task is to identify and predict user intent so that CAs can process the information accordingly to satisfy the users' information needs. However, utterances which were labeled *Greetings/Gratitude*, *Junk*, and *Others* do not contribute to the purpose of providing information about QA related user intent. Therefore, we remove occurrences of these labels. Note that we only remove these labels if there are more than one label of the given utterance. For example, if the annotation for the given utterance is *GG+OQ*, we transform the annotation into *OQ*. If the annotation is just *GG*, no transformation is needed. This reduces the number of unique label combinations from 316 to 152.

In addition, some label combinations of user intent labels are quite rare in the data. As indicated in Figure 1a, the top frequent label combinations have hundreds of occurrences in the data (e.g. *PA*, *OQ*, *PF*, *FD+PA*, *FD*), while the least frequent labels only have exactly one occurrence (e.g. *CQ+FD+IR+RQ*, *CQ+FD+FQ+PF*). These rare label combinations are very likely due to minor annotation errors or noise with MTurk. Annotation quality assurance was performed based on the dialog-level inter-rater agreement [19] to keep the complete dialog intact and thus may result in minor noise on an utterance level. We also plot the cumulative distribution of label combinations for better illustration in Figure 1b. The most frequent 32 label combinations constitute 90% of total label combination occurrences as marked in the figure. All 12 user intent labels are individually present in these 32 most frequent combinations except for *Others*. For the rest of the label combinations, we randomly sample one of the labels from each combination as the user intent label for the given utterance. For example, if the annotation for the given utterance is *CQ+FD+IR+RQ*, we transform it into a single label by randomly sampling one of the four labels, such as *CQ*. Therefore, the total number of label combinations in the data was reduced to

33 (including *Others*). We adopted this setting since these rare label combinations are very likely due to minor annotation errors. In addition, it would be very difficult to learn a prediction model for these label combinations with few instances.



(a) Label combination occurrence from the most frequent to the least frequent.

(b) Cumulative occurrence probability added from the most frequent to the least frequent. The marked point is (32, 0.9).

Figure 1: Label combination distribution

For UDC, we observe a similar label combination distribution. So we preprocess the data in the same way. We have 34 label combinations for the UDC with 27 of them overlapping with MSDialog.

4 BASELINES AND FEATURE ANALYSIS

In this section, we extract several features following previous work [2, 5] and adopt different ML methods to build baseline models. In addition to reporting baseline performance, we also perform feature importance analysis to identify key factors in user intent prediction.

4.1 Features

We extract three groups of features to detect user intent in information-seeking conversations, including content, structural, and sentiment features. An overview of the features is provided in Table 3. Although MSDialog is derived from forum data, it is considered as a dialog dataset [19]. Thus, we refrain from developing features that can only be extracted from the metadata of the forum, such as user authority level or answer votes, so that our method can be applicable to dialog systems. The features are designed to capture the content and sentiment characteristics of the utterances as well as the structural information of the dialogs.

Content features. We build a TF-IDF representation of utterances and compute the cosine similarity of the given utterance with the dialog initial utterance (which typically is the question that initiates the QA dialog), and the entire dialog. These features are meant to capture the relevance level of the given utterance to the dialog in a general way. In addition, the presence of question marks is a strong indicator that the current utterance contains a question. Moreover, we assume that 5W1H keywords (what, where, when, why, who, and how) can suggest the type of the question.

Structural features. The position of an utterance in a dialog can reveal crucial information about user intent. Intuitively, answers tend to be at even number positions in a dialog, while user feedback and follow up questions tend to be at odd number positions. In addition, we include the utterance length with and without duplication removal and stemming. We analyzed the data and found that utterances containing positive feedback are relatively short, while utterances containing questions or answers tend to be long as they typically contain details. Finally, if the given utterance is generated by the information seeker (dialog starter), it is more likely to contain user related questions or user feedback. The structural features not only provide individual characteristics for utterances, but also evaluate the utterances on a dialog level.

Sentiment features. We expect sentiment features to be useful in identifying user feedback and gratitude expressions. In information-seeking conversations, positive and negative sentiments do not necessarily determine the feedback type. However, we expect them to be correlated to some extent. We also include classic indicators of sentiment, such as the presence of "thank", "does not/did not" and exclamation marks. In addition, we use VADER [8] to compute the positive/negative/neutral sentiment scores. We also count the number of positive and negative words using an opinion lexicon [12].

4.2 Methods and Evaluation Metrics

4.2.1 Methods. For each utterance, we extract a set of features as described in Section 4.1. To apply traditional ML methods with features to this task, we need to transform this multi-label classification problem to multi-class classification. Three transformation strategies are typically used: binary relevance, classifier chains, and label powerset. Binary relevance does not consider the label correlations and label powerset generates new labels for every label combination. So we choose classifier chains as the transformation strategy for traditional ML methods. This strategy performs binary classifications for each label and take predictions for previous labels as extra features. This transformation strategy is the best fit for our task as it considers the label dependency without explicitly generating new labels for every label combination. We adopt classic ML methods, including Naive Bayes classifier, SVM, random forest, and AdaBoost as baseline classifiers. In addition, we use ML-kNN, which supports multi-label classification by nature.

4.2.2 Metrics. Due to the nature of the intent prediction task, we adopt metrics suitable for multi-label classification problems.

Accuracy (Acc). It is known as the intersection over the union (IoU) in the multi-label classification settings. Accuracy is defined as the number of correctly predicted labels divided by the union of predicted and true labels for every utterance. For example, if the model predicts "*PA+CQ*" for the given utterance while the ground truth is "*PA+IR*", then the accuracy is $\frac{1}{3}$. The reported performance is the average metric over all utterances.

Precision, recall and F1 score. Precision is defined as the number of correctly predicted labels divided by predicted labels. Recall is defined as the number of correctly predicted labels divided by true labels. F1 is their harmonic mean. These metrics provide an overall performance evaluation for all utterances.

4.3 Main Experiments and Results

4.3.1 Experimental Setup. We split the labeled subset of MSDialog into training, validation, and test sets. Table 4 gives the statistics of the three sets. We have to point out that although this dataset is not that large, the annotation cost for such fine-grained user intent in conversations is quite high (estimated around $1,700 on MTurk according to [19]). This dataset size is also larger than the data used in related work [2, 23]. The models are trained with

Table 3: Features extracted for user intent prediction in information-seeking conversations.

Feature Name	Group	Description	Type
Initial Utterance Similarity	Content	Cosine similarity between the utterance and the first utterance of the dialog	Real
Dialog Similarity	Content	Cosine similarity between the utterance and the entire dialog	Real
Question Mark	Content	Does the utterance contain a *question mark*	Binary
Duplicate	Content	Does the utterance contain the keywords *same, similar*	Binary
5W1H	Content	Does the utterance contain the keywords *what, where, when, why, who, how*	One-hot vector
Absolute Position	Structural	Absolute position of an utterance in the dialog	Numerical
Normalized Position	Structural	Normalized position of an utterance in the dialog (AbsPos divided by # utterances)	Real
Utterance Length	Structural	Total number of words in an utterance after stop words removal	Numerical
Utterance Length Unique	Structural	Unique number of words in an utterance after stop words removal	Numerical
Utterance Length Stemmed Unique	Structural	Unique number of words in an utterance after stop words removal and stemming	Numerical
Is Starter	Structural	Is the utterance made by the dialog starter	Binary
Thank	Sentiment	Does the utterance contain the keyword *thank*	Binary
Exclamation Mark	Sentiment	Does the utterance contain an *exclamation mark*	Binary
Feedback	Sentiment	Does the utterance contain the keyword *did not, does not*	Binary
Sentiment Scores	Sentiment	Sentiment scores of the utterance computed by VADER [8] (positive, neutral, and negative)	Real
Opinion Lexicon	Sentiment	Number of positive and negative words from an opinion lexicon	Numerical

scikit-multilearn[9] and scikit-learn[10] on the training set. We tune the hyper-parameters on the validation set based on accuracy and report the performance on the test set.

Table 4: Statistics of training, validation, and testing sets

Item	Train	Val	Test
# Utterances	8,064	986	970
Min. # Turns Per Dialog	3	3	3
Max. # Turns Per Dialog	10	10	10
Avg. # Turns Per Dialog	4.58	4.48	4.43
Avg. # Words Per Utterance	70.42	67.53	68.64

4.3.2 Baseline Results. The baseline results are presented in Table 5. Two ensemble methods, random forest and AdaBoost achieve the best overall performance of all baseline classifiers. AdaBoost achieves the best accuracy while random forest achieves the best F1 score. Surprisingly, ML-kNN performs relatively poorly despite its nature of an adapted algorithm for multi-label classification.

Table 5: Experiment results for baseline classifiers

Methods	Acc	Precision	Recall	F1
ML-kNN	0.4715	0.6322	0.4471	0.5238
NaiveBayes	0.4870	0.5563	0.4988	0.5260
SVM	0.6342	0.7270	0.5847	0.6481
RandForest	0.6268	**0.7657**	0.5903	**0.6667**
AdaBoost	**0.6399**	0.7247	**0.6030**	0.6583

4.4 Additional Feature Importance Analysis

4.4.1 Feature Group Analysis. We use one of the best baseline classifiers, random forest, and different combinations of feature groups to analyze the feature importance on a group level. The hyper-parameters are set to the best ones tuned on all features .

For using a single feature group, structural features is the most important feature group as presented in Table 6. Structural features and content features are significantly more important than sentiment features. We expect the sentiment features to capture the sentiment in user feedback but they might not be able to effectively discriminate other user intent. Structural features provide better performance than content features. We believe that this can be explained by the fact that hand-crafted content features cannot capture the complex user intent dynamics in human information-seeking conversations.

Table 6: Experiment results for different feature groups

Group(s)	Acc	Precision	Recall	F1
Content	0.5272	0.6097	0.4821	0.5384
Structural	**0.5809**	**0.6871**	**0.5434**	**0.6068**
Sentiment	0.3306	0.4087	0.3222	0.3603
Con+Str	0.6081	0.7393	0.5640	0.6399
Con+Sen	0.5577	0.6523	0.5179	0.5774
Str+Sent	**0.6110**	**0.7569**	**0.5672**	**0.6485**
All	**0.6268**	**0.7657**	**0.5903**	**0.6667**

For combinations of two feature groups, content+structural features and structural+sentiment features achieve comparable results. However, structural+sentiment features achieve slightly higher results on all metrics. The performance of using two groups of features is higher than using one of these two feature groups individually. Thus, combining structural features with another feature group boosts the performance of using structural features alone. Interestingly, content+sentiment features is unable to outperform the structural features alone. The results of using all features is the highest among all settings, confirming that all feature groups contribute to the performance of user intent prediction.

4.4.2 Feature Importance Scores. In the previous section, we evaluated the feature importance on a group level. In this section we focus on individual features to provide a more fine-grained analysis. We use random forest to output individual feature importance scores.[11] As described in Section 4.2, we used classifier chains to transform this multi-label classification problem. This method expands the feature space by including previous label predictions as new features for the current label prediction. This makes it not appropriate to evaluate original features. Thus, we use the Label Powerset method as the data transformation strategy for this section. The relative feature importance scores are presented in Table 7. This analysis can identify key factors in user intent prediction.

We summarize our observations as follows: (1) Structural features including "Absolute Position", "Normalized Position", "Is Starter" are ranked in the top-5 in terms of feature importance. Moreover,

[9] http://scikit.ml/ [10] http://scikit-learn.org/stable/

[11] https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

Table 7: Individual feature importance from a random forest classifier with relative importance scores. "Str", "Con", "Sen" refer to "Structural", "Content", "Sentiment" respectively.

Rank	Feature	Group	Impt	Rank	Feature	Group	Impt
1	AbsPos	Str	1.0	13	Lex(Pos)	Sen	0.2814
2	InitSim	Con	0.9745	14	Lex(Neg)	Sen	0.2337
3	NormPos	Str	0.8684	15	Thank	Sen	0.1607
4	Starter	Str	0.8677	16	How	Con	0.08074
5	DlgSim	Con	0.6778	17	Dup	Con	0.06908
6	SenScr(Neu)	Sen	0.6465	18	What	Con	0.06576
7	SenScr(Pos)	Sen	0.5601	19	ExMark	Sen	0.06424
8	Len	Str	0.5335	20	When	Con	0.05989
9	LenUni	Str	0.4381	21	Feedback	Sen	0.02859
10	LenStem	Str	0.4354	22	Where	Con	0.02356
11	SenScr(Neg)	Sen	0.3495	23	Why	Con	0.0232
12	QuestMark	Con	0.3003	24	Who	Con	0.01423

other structural features, such as various forms of utterance length are observed to be relatively informative in general. This confirms the results in Section 4.4.1 that the structural feature group is the most important one. (2) "Initial Utterance Similarity" and "Dialog Similarity" are content features that can be highly informative for identifying user intent. Both features are indicators of how closely the utterance connects with the information-seeking process. Other content features, such as "5W1H", however, contribute little to predicting user intent. (3) Some sentiment features are relatively important in identifying user intent, such as positive and neutral sentiment scores. However, some other sentiment features contribute little to the task, such as the existence of exclamation marks and "thank". (4) We observe that features ranked from the 15^{th} to the last one in Table 7 are all "keyword features". These features are based on a simple rule that whether the given utterance contains pre-defined keywords. For example, the "5W1H" feature looks for "what/where/when/why/who/how" in the given utterance and the "Feedback" feature looks for "did not/does not". The major drawback of manual feature engineering is amplified in this task due to the complexity and diversity of human information-seeking conversations.

5 ENHANCED NEURAL CLASSIFIERS

We expected the content of an utterance to be a good indicator of user intent types compared to other features. However, as shown in Section 4.4, the hand-crafted content features are unable to capture the complex characteristics of human information-seeking conversations. Thus, in this section we adopt neural architectures to automatically learn representations of utterances without feature engineering.

5.1 Our Approach

5.1.1 Base Models. Given the previous success in modeling text sequences using CNN and bidirectional LSTM (BiLSTM) [7], we choose these two architectures as our base models. Although utterances are grouped as dialogs, the base models consider utterances independently.

Given an utterance $u_i^k = \{w_1, w_2, \ldots, w_m\}$ (the k-th utterance in the i-th dialog) that contains m tokens, we first transform the sequence of tokens into a sequence of token indices $S = \{s_1, s_2, \ldots, s_m\}$. Then we pad the sequence S to a fixed length n (the max sequence length). Both CNN and BiLSTM start with an embedding layer initiated with pre-trained word embeddings. Preliminary experiments indicated that using MSDialog (complete set) to train word embeddings is more effective than using GloVe [18] in terms of final model performance. The embedding layer maps each token in the utterances to a word embedding vector with a dimension of d.

We focus on the CNN model following previous work [10] here, because it achieves better performance in our experiments. After the embedding layer, filters with the shape (f, d) are applied to a window of f words. f is also referred to as the filter size. Concretely, a convolution operation is denoted as

$$c_i = \sigma(\mathbf{w} \cdot \mathbf{e}_{i:i+f-1} + b) \tag{1}$$

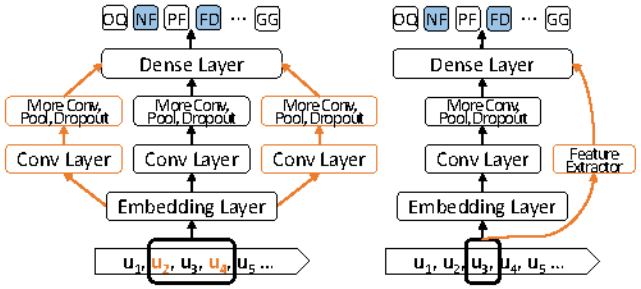
Where c_i is the feature generated by the i-th filter with weights $\mathbf{w}$ and bias b. This filter is applied to an embedding matrix, which is the concatenation from the i-th to the $(i + f - 1)$-th embedding vectors. An non-linearity function (ReLU) is also applied. This operation is applied to every possible window of words and generates a feature map $\mathbf{c} = \{c_1, c_2, \ldots, c_{n-f+1}\}$. More filters are applied to extract features of the utterance content. Max pooling are applied to select the most salient feature of a window of f' features by taking the maximum value $\hat{c}_i = max\{\mathbf{c}_{i:i+f'-1}\}$. f' denotes the max pooling kernel size. A dropout layer is applied after the pooling layer for regularization.

After the last convolutional layer, we perform global max pooling by taking the maximum value $\hat{c} = max\{\mathbf{c}\}$ for the feature map $\mathbf{c}$ at this step. This operation reduces the dimension of the tensor to one. This tensor is further transformed to an output tensor of shape $(1, l)$, where l is the number of user intent labels (12 for our task). Sigmoid activation is applied to each value of the output tensor to squash the value to a confidence level between 0 and 1. A threshold is chosen to determine whether the given label present in the final prediction. If the model is not confident of predicting any label, the label of the highest confidence level is the prediction. We tuned the threshold with the validation data.

5.1.2 Incorporate Context Information. As shown in the previous work [19], user intent follows clear flow patterns in information-seeking conversations. The user intent of a given utterance is closely related to the utterances around it, which compose the *context* for the given utterance. Incorporating context information into neural models is easier compared to that for traditional ML methods shown in Section 4. We consider two ways as follows.

Direct Expansion. The most straightforward way to incorporate context information is to expand the given utterance with its context. Concretely, the expanded utterance for u_i^k is $\hat{u}_i^k = u_i^{k-1} \oplus u_i^k \oplus u_i^{k+1}$, where $\oplus$ is the concatenate operator. $\hat{u}_i^k$ is considered as the given utterance in base models.

Context Representation. Given an expanded utterance $\hat{u}_i^k$ as input, the neural architecture first segments it into three original utterances of u_i^{k-1}, u_i^k, and u_i^{k+1}. We apply convolution operations and max pooling to the utterances separately as shown in Figure 2a. After global pooling following the last convolutional layer, the three one-dimensional tensors are concatenated for final predictions. This approach extracts features from the given utterance and its context separately. Thus, we are able to learn the importance of the given utterance and its context by tuning context-specific hyperparameters, such as the number of filters for context utterances.



(a) CNN-Context-Rep (b) CNN-Feature

Figure 2: Architectures for enhanced neural classifiers. Components marked orange are extra information incorporated into the base CNN model (black). The utterance in bold is the current utterance. Predicted labels are marked blue.

5.1.3 Incorporate Extra Features. We found many useful features for user intent prediction such as structural features in the feature importance analysis in Section 4.4. However, nearly half of the features can not be exploited by only looking at a single utterance. For example, normalized/absolute utterance position and utterance similarity with the dialog/initial utterance cannot be captured without a holistic view over the entire dialog. Some of the uncaptured features are highly informative. This motivates us to incorporate hand-crafted features into the neural architectures. As shown in Figure 2b, all hand-crafted features described in Section 4.1 are incorporated into neural architectures at the last dense layer. The feature vector is concatenated with the neural representation of the utterance before making final predictions.

Finally, we combine two base models with various extra components to produce several systems for comparison as follows:

- **CNN**. The base CNN model that consists of three convolutional layers with the same filter size.
- **CNN-Feature**. The CNN model that incorporates extra hand-crafted features at the last layer.
- **CNN-Context**. The CNN model that incorporates context information with direct expansion.
- **CNN-Context-Rep**. The CNN model that incorporates context information with context representation.
- **BiLSTM**. The BiLSTM model that represents the given utterance both in the ordinary order and the reverse order.
- **BiLSTM-Context**. The BiLSTM model that incorporates context information with direct expansion.

5.2 Experiments and Evaluation

5.2.1 Neural Baselines. In addition to the base model of **BiLSTM** and **CNN**, we further introduce two commonly used neural models for text classification as baselines. For both new baselines, we modify the models to generate multi-label predictions.

CNN-MFS. The CNN model with multiple filter sizes as described by Kim [10] is a pioneer model to apply neural networks to text classification. This model uses different filter sizes of 3, 4, and 5 to generate feature maps of different window sizes.

Char-CNN. Zhang et al. [32] introduced a character-level CNN for text classification. There are two variants of the model, a large one and a small one, depending on the numbers of convolutional filters and dense layer units. We report the performance on the small model as it achieves better results in our task.

5.2.2 Experimental Setup. We use the same data and metrics as in baseline experiments in Section 4. All models are implemented with TensorFlow[12] and Keras[13]. Hyper-parameters are tuned with the validation data. We found that setting (convolutional filters, dropout rate, dense layer units, max sequence length, convolutional filters for context, and dense layer units for context) to (1024, 0.6, 256, 800, 128, 128) respectively turned out to be the best setting for our best performing model CNN-Conext-Rep. The convolutional filter size and pooling size are set to (3, 3). All models are trained with a NVIDIA Titan X GPU using Adam [11]. The initial learning rate is 0.001. The parameters of Adam, β_1 and β_2 are 0.9 and 0.999 respectively. The batch size is 128. For the word embedding layer, we trained word embeddings with Gensim[14] with CBOW model using MSDialog (complete set). The dimension of word embedding is 100. Word vectors are set to trainable.

5.2.3 Evaluation Results. We select the two strongest feature based classifiers from Section 4 as feature based baselines in addition to neural baselines. They are random forest with the best F1 score and AdaBoost with the best accuracy. The performance comparison of models is presented in Table 8.

Table 8: Results comparison. The significance test can only be performed on accuracy. In a multi-label classification setting, accuracy gives a score for each individual sample, while other metrics evaluate the performance over all samples. ‡ means statistically significant difference over the best baseline with $p < 10^{-4}$ measured by the Student's paired t-test.

Method Types	Methods	Accuracy	Precision	Recall	F1
Feature based Baselines	Random Forest	0.6268	**0.7657**	0.5903	**0.6667**
	AdaBoost	**0.6399**	0.7247	**0.6030**	0.6583
Neural Baselines	BiLSTM	0.5515	0.6284	0.5274	0.5735
	CNN	**0.6364**	0.7152	**0.6054**	**0.6558**
	CNN-MFS	0.6342	**0.7308**	0.5919	0.6541
	Char-CNN	0.5419	0.6350	0.4940	0.5557
Neural Classifiers	BiLSTM-Context	0.6006	0.6951	0.5640	0.6227
	CNN-Feature	0.6509	0.7619	0.6110	0.6781
	CNN-Context	0.6555	0.7577	0.6070	0.6740
	CNN-Context-Rep	**0.6885**‡	**0.7883**	**0.6516**	**0.7134**

The base CNN model without feature engineering achieves similar results with the strongest feature based baselines. The performance of the base CNN model is better than the base BiLSTM model. CNN-MFS takes advantage of different filter sizes and also achieves comparable results. Char-CNN, however, performs poorly in this task. Char-CNN does not use pre-trained word embeddings because it learns features from a character-level which would require much more training data.

BiLSTM performs poorly in this task. Even though BiLSTM-Context has a major improvement over BiLSTM, it has inferior results compared to the base CNN model. Compared to non-factoid question answering or chatting, utterances in information-seeking conversations tend to be longer. BiLSTM(-Context) tries to model a

[12] https://www.tensorflow.org/ [13] https://keras.io/
[14] https://radimrehurek.com/gensim/

holistic sequence dependency and thus performs poorly on handling these long utterances.

The best result of the feature based baselines is slightly higher than neural baselines. This can be accounted for by the lack of information in neural models. Even though we assume that most of the content and sentiment features can be learned by neural models, the neural models have no access to most of the structural features. Thus, we incorporate all the features to neural models to produce CNN-Feature model. This model outperforms all baseline classifiers. This confirms that incorporating dialog-level information can be beneficial to predicting user intent.

Both CNN-Context and CNN-Context-Rep outperform baseline models and CNN-Feature without explicit feature engineering. These results demonstrate the effectiveness of the implicit feature learning of neural architectures. CNN-Context-Rep performs better than CNN-Context. This indicates that incorporating high-level features of context information learned by neural architectures is better than directly capturing the raw context information. In a multi-label classification setting, accuracy produces a score for each individual sample, while precision/recall/F1 evaluate the performance over all samples. Thus, accuracy is the only metric that is suitable for significance tests. Our best model, CNN-Context-Rep achieves statistically significant improvement over the best baseline with $p < 10^{-4}$ measured by the Student's paired t-test.

5.3 Generalization on Ubuntu Dialogs

In this section, we would like to evaluate the generalization performances of different methods on other data in addition to MSDialog. We train different models on MSDialog and test them on the Ubuntu Dialog Corpus (UDC). We select the two best performing feature based classifiers (random forest and AdaBoost) and the best neural model (CNN-Context-Rep) to test the generalization performance. Although the number of annotated Ubuntu dialogs is limited, it is sufficient to demonstrate the predicting performance. We split the annotated UDC data into validation and test sets with an equal size. We train the model on MSDialog data only and tune the hyper-parameters on the UDC validation set. The performance on the UDC test set is presented in Table 9.

Table 9: Testing performance on UDC of different models trained with MSDialog. The significance test can only be performed on accuracy. ‡ means statistically significant difference over both strongest feature based baselines with $p < 0.01$ measured by the Student's paired t-test.

Methods	Accuracy	Precision	Recall	F1
Random Forest	0.4405	**0.6781**	0.4077	0.5092
AdaBoost	0.4430	0.5913	0.4187	0.4902
CNN-Context-Rep	**0.4708**‡	0.5647	**0.5129**	**0.5375**

The generalization results on UDC are not as good as MSDialog. Although MSDialog and UDC both consist of multi-turn information-seeking dialogs from the technical support domain, the drastically different language style adds difficulty for model generalization and transferring. In this challenging setting, CNN-Context-Rep still achieves statistically significant improvement over both baselines in terms of accuracy with $p < 0.01$ measured by the Student's paired t-test.

5.4 Hyper-parameter Sensitivity Analysis

We further analyze the impact of two hyper-parameters on CNN-Context-Rep: the number of convolutional filters for the given utterance and the max sequence length. The choices for number of convolutional filters are (64, 128, 256, 512, 1024). We tune the max sequence length in (50, 100, 200, …, 1000). As presented in Figure 3, the performance gradually increases as the number of filters increases. The best performance is at 1,024 filters. This confirms our expectation that more convolutional filters can extract richer features and thus produce better results. In addition, the performance fluctuates as the max sequence length increases. Performance with larger ($\geqslant 800$) max sequence length are better in general.

5.5 Case Study

Table 10 gives examples of the predictions that different systems fail to make. In the first utterance, the agent asks for the user's iOS version before providing a potential answer, which is a very common pattern of agents' responses. Our CNN-Context-Rep is able to identify the *Information Request* in the utterance while AdaBoost cannot. In the second utterance, both models fail to predict the *Negative Feedback*. This might be due to the fact that the feedback is not explicitly expressed. In addition, it could be relevant that the number of feedback utterances in the training data is relatively limited compared to questions and answers, which makes it more difficult to predict positive/negative feedback.

Table 10: Two utterances with their ground-truth and predicted user intent labels. Bold font indicates mispredicted content or labels. "Ours" refers to CNN-Context-Rep.

Hello. Welcome to Skype Community! **Please provide us the iOS version of your iPad**. The required iOS version for iPad is iOS 8 or higher and for the new Skype on iOS requires iOS 9 or higher. For more information, click here. Hope this helps. Let me know if you need further assistance. Thank you!			
Ground truth: **IR**, PA	Ours: **IR**, PA	AdaBoost: PA	Actor: agent
After modified the Windows entry, value of regedit, **the error also happened**. When I use C++ for creating another new Microsoft::Office::Interop::PowerPoint:: Application instance, the COMException is throwed.			
Ground truth: FD, **NF**	Ours: FD	AdaBoost: FD	Actor: user

6 CONCLUSIONS

In this paper, we studied two approaches to predict user intent in information-seeking conversations. First we use different ML methods with a rich feature set, including the content, structural, and sentiment features. We perform thorough feature importance analysis on both group level and individual level, which shows that structural features contribute most in this prediction task. Given findings from feature analysis, we construct enhanced neural classifiers to incorporate context information for user intent prediction. The enhanced neural model without feature engineering outperforms the baseline models by a large margin. Our findings can provide insights in the important factors of user intent prediction in information-seeking conversations. Future work will consider other methods for user intent prediction. Utilizing the predicted user intent to rank or generate conversation responses in an information-seeking setting is also interesting to explore.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] N. J. Belkin, C. Cool, A. Stein, and U. Thiel. Cases, Scripts, and Information-Seeking Strategies: On the Design of Interactive Information Retrieval Systems. 1970.
[2] S. Bhatia, P. Biyani, and P. Mitra. Classifying User Messages For Managing Web Forum Data. In *WebDB '12*, 2012.
[3] W. B. Croft and R. H. Thompson. I3R: A New Approach to the Design of Document Retrieval Systems. *J. Am. Soc. Inf. Sci.*, 1987.
[4] D. Datta, V. Brashers, J. Owen, C. White, and L. E. Barnes. A Deep Learning Methodology for Semantic Utterance Classification in Virtual Human Dialogue Systems. In *IVA '16*, 2016.
[5] S. Ding, G. Cong, C. Lin, and X. Zhu. Using Conditional Random Fields to Extract Contexts and Answers of Questions from Online Forums. In *ACL '08*, 2008.
[6] A. C. Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, and R. Kreuz. AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, 2001.
[7] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 2005.
[8] C. J. Hutto and E. Gilbert. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *ICWSM '14*, 2014.
[9] H. Khanpour, N. Guntakandla, and R. D. Nielsen. Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. In *COLING '16*, 2016.
[10] Y. Kim. Convolutional Neural Networks for Sentence Classification. In *EMNLP '14*, 2014.
[11] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR '15*, 2015.
[12] B. Liu, M. Hu, and J. Cheng. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *WWW '05*, 2005.
[13] Y. Liu, K. Han, Z. Tan, and Y. Lei. "Using Context Information for Dialog Act Classification in DNN Framework". In *EMNLP '17*, 2017.
[14] R. Lowe, N. Pow, I. Serban, and J. Pineau. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *SIGDIAL '15*, 2015.
[15] G. Marchionini. Exploratory Search: From Finding to Understanding. *Commun. ACM*, 2006.
[16] R. N. Oddy. *Information Retrieval through Man-Machine Dialogue*. 1977.
[17] A. Olney, M. Louwerse, E. Matthews, J. Marineau, H. Hite-Mitchell, and A. Graesser. Utterance Classification in AutoTutor. In *HLT-NAACL-EDUC*, 2003.
[18] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *EMNLP '14*, 2014.
[19] C. Qu, L. Yang, W. B. Croft, J. R. Trippas, Y. Zhang, and M. Qiu. Analyzing and Characterizing User Intent in Information-seeking Conversations. In *SIGIR '18*, 2018.
[20] F. Radlinski and N. Craswell. A Theoretical Framework for Conversational Search. In *CHIIR '17*, 2017.
[21] S. Shiga, H. Joho, R. Blanco, J. R. Trippas, and M. Sanderson. Modelling information needs in collaborative search conversations. In *SIGIR '17*, 2017.
[22] A. Stolcke, N. Coccaro, R. Bates, C. Van Ess-Dykema, C. Van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 2000.
[23] D. Surendran and G. A. Levow. Dialog act tagging with support vector machines and hidden markov models. In *ICSLP '06*, 2006.
[24] J. R. Trippas, D. Spina, L. Cavedon, H. Joho, and M. Sanderson. Informing the Design of Spoken Conversational Search: Perspective Paper. In *CHIIR '18*, 2018.
[25] R. W. White and R. A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm*. Synthesis Lectures on Information Concepts, Retrieval, and Services. 2009.
[26] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *ACL '17*, 2017.
[27] Z. Yan, N. Duan, P. Chen, M. Zhou, J. Zhou, and Z. Li. Building Task-Oriented Dialogue Systems for Online Shopping. In *AAAI '17*, 2017.
[28] L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, and H. Chen. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In *SIGIR '18*, 2018.
[29] Y. Yang, W. T. Yih, and C. Meek. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *EMNLP '15*, 2015.
[30] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li. Neural Generative Question Answering. In *IJCAI '15*, 2015.
[31] X. Zhang and H. Wang. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI '16*, 2016.
[32] X. Zhang, J. J. Zhao, and L LeCun. Character-level Convolutional Networks for Text Classification. In *NIPS '15*, 2015.
[33] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft. Towards conversational search and recommendation: System ask, user respond. In *CIKM '18*, 2018.

A ADDITIONAL FIGURE

We include an additional figure to illustrate our findings from above. Figure 3 shows the results of the hyper-parameter sensitivity analysis.

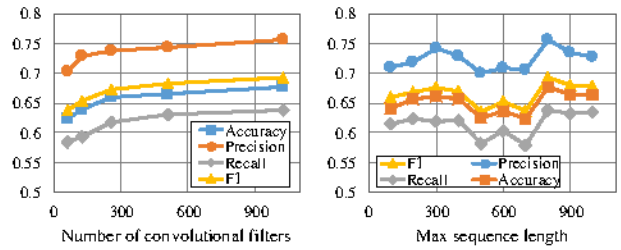


Figure 3: Performance of CNN-Context-Rep with different number of convolutional filters and max sequence length.