Submission Template for ACM Papers

This is Short Title of the paper, used in page headers

This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word

Bas de Blok

b.deblok@students.uu.nl,

Mykola Chuprynskyy

m.chuprynskyy@students.uu.nl,

Melissa Rueca

m.rueca@students.uu.nl, 8463603

Luka van Hoeven

l.vanhoeven@students.uu.nl,

Dialog systems are machine-based systems designed to help users to solve specific tasks. The goal of this project was to design, develop and implement a dialog system capable of interpreting user preferences such as cuisine type, location and price range and providing restaurant recommendations. We implemented two baseline systems: majority-class baseline and rule-based baseline. We also trained two machine learning classifiers: logistic regression and feed forward neaural network. We evaluated the model using accuracy and we conducted an error analysis to identify challenging dialog act and difficult utterances.   
ADD THE PART 1b

1. Introduction

Dialog systems are machine-based systems designed to help users to solve specific tasks. They are commonly used in application such as controlling appliances, finding restaurants or making travel reservations.

The main goal of the project was to develop a dialog system able to provide restaurant recommendations. The system interprets user preferences such as type of cuisine, location and price range and provides appropriate recommendations or ask appropriate questions. Users interact with the system by asking questions, providing information or responding to the system’s questions.

We used rule-based methods and machine learning techniques. The system includes a dialog manager, two baseline models for comparison and two machine learning classifiers (logistic regression model and feed forward neural network) to assign the correct dialog act to each utterance.

This paper describes the dataset and preprocessing steps (Section 2), the implementation of baseline and machine learning system and relative evaluation (Section 3), the dialog manager (Section 4), the reasoning (Section 5) and the configurability (Section 6).

1. DATA

The dataset used in the project is derived from the second Dialog State Tracking Challenge, which focuses on restaurant recommendations dialogs.

* 1. Dataset

The dataset contains 3235 dialogs, each representing an interaction between a user and the system. The user expresses preferences such as type of cuisine, location and price range and the system responds either by giving recommendations, requesting additional preference or giving general communicative statements.

Analyzing the dataset, it can be observed that there are some utterances repeated across dialogs, accounting for approximately xx% (INSERT REAL NUMBER) of the total.

The dialogs were collected using automatic speech recognition, which can introduce transcription errors or linguistic inconsistencies. Specifically, we may encounter misspelled words, onomatopoeic expression representing sounds or noises, missing punctuation or colloquial expressions typical of spoken language. These issues can affect system performance, for example a misspelled word might not be classified correctly. Therefore, it was necessary to handle these cases to ensure proper recognition, for example the Levenshtein edit distance was used to handle misspellings.

The dataset includes 15 dialog acts to classify user and system the utterances: ack, affirm, bye, confirm, deny, hello, inform, negate, null, repeat, reqalts, reqmore, request, restart, thankyou.

* 1. State Transition Diagram

We modelled the dialog system as a state transition diagram, in which each state is a stage of the interaction.

ADD IMAGE OF STATE TRANSITION DIAGRAM (how? Too big) !!!

The state transition diagram contains 7 states:

* Welcome
* Ask area
* Ask type of food
* Ask for price range
* Give choice to pick the suggestion or restart conversation
* End conversation

The system transitions between the states depends on user inputs and slot fulfilment.

Here an example dialog snippet:

Task 01527: You want to find a moderately priced restaurant and it should serve bistro food. If there is no such venue how about british type of food. Make sure you get the phone number.

turn index: 0

system: Hello , welcome to the Cambridge restaurant system? You can ask for restaurants by area , price range or food type . How may I help you?

user: im looking for a moderately priced restaurant that serves

speech act: inform(type=restaurant,pricerange=moderate,task=find)

turn index: 1

system: What part of town do you have in mind?

user: any part of town

speech act: inform(area=dontcare)

turn index: 2

system: What kind of food would you like?

user: bistro food

speech act: inform(food=bistro)

turn index: 3

system: I'm sorry but there is no restaurant serving bistro food

user: is there a moderately priced restaurant that serves british food

speech act: confirm(pricerange=moderate,type=restaurant,food=british)

turn index: 4

system: You are looking for a british restaurant right?

user: yes

speech act: affirm()

turn index: 5

system: saint johns chop house serves british food in the moderate price range

user: could i get their phone number

speech act: request(phone)

turn index: 6

system: The phone number of saint johns chop house is 01223 353110 .

user: thank you good bye

speech act: thankyou()|bye()

In this example the first system utterance corresponds to the “Welcome” state. The system enters the “Ask area” state when it asks “What part of town do you have in mind?” and the “Ask type of food” state when the system asks “What kind of food would you like?”. The user’s confirmation and the system providing the restaurant recommendation shows the “Give choice to pick the suggestion or restart conversation” state. The final “thank you good bye” corresponds to the “End conversation” state.

1. MACHINE LEARNING

Now we explain the preprocessing of data, the implementation of the two baseline-systems, the machine learning classifiers used and the evaluation of the systems.

* 1. Data preprocessing

Data preprocessing was a necessary step to prepare the dataset for all subsequent phases of the project.

The dataset was first subjected to a text cleaning, where all the utterances were normalized by converting them to lowercase and removing punctuation. This made the text more consistent and easier to analyse during classification.

Furthermore, two distinct versions of the dataset were constructed: in addition to the original dataset, which includes duplicate utterances, a deduplicated version was created, keeping only one instance of each utterance. Both the original and deduplicated datasets were then split into a training set (85% of the data) and a test set (15% of the data).

INSERT STATISTICS ON THE DATA!!

* 1. Baseline systems

We implemented two baseline systems.

The first one was a majority-class baseline, which assigns the most frequent dialog act to every utterance, regardless of its content. In our dataset, the most frequent dialog act is inform, which is almost 40% of all utterances in the deduplicated dataset.

The second one was a rule-based system, which uses keyword matching to assign a dialog act to each utterance. An order of priority was defined among the dialog acts so that more distinctive acts are considered first, preventing them from being overshadowed by more common acts. Based on this priority order, once a matching dialog act was identified for an utterance, it was immediately assigned to that utterance. INSERT REAL NUMBER!!

* 1. Machine learning classifiers

For the machine learning part, we decided to train two different machine learning classifiers on the dialog act data: Logistic regression and Feed Forward neural network.

The Logistic Regression model uses TF-IDF features extracted from the utterances and it can also include character n-grams to capture patterns inside words. The model learns to predict the dialog act of each utterance from the training data and is evaluated on the test set using accuracy, macro F1-score and a confusion matrix.

The Feed Forward neural network uses a bag of words representation of the utterance and learns to predict the dialog act from the training data.

INSERT REAL NUMBER!!

* 1. Evaluation

After training the models, we evaluated them on both the original and deduplicated test tests.

We evaluated the models’ performance using accuracy on the test sets. Separate functions were implemented to compute accuracy for each model (rule-based baseline, majority-class baseline, feed forward neural network and logistic regression model). In this way we provided a global measure of performance.

Beyond accuracy, we analyzed the utterances that were misclassified to identify patterns causing errors. For each model, we calculated the F1-score for each dialog act and highlighted the most difficult acts to classify. This allowed us to see if certain dialog acts were consistently mispredicted and whether improvements in preprocessing or other steps could reduce these errors.

We also analysed difficult instances by writing our test cases. There were of two types: utterances affected by speech recognition error or misspellings (for example: “can you find an italian restaurant”) and utterances containing negation (for example: “i don’t want italian restaurant”). The system often failed to predict the correct dialog act for these utterances and this highlights the areas where robustness could be improved.

INSERT REAL NUMBER!!!

1. DIALOG MANAGER
2. REASONING
3. CONFIGURABILITY
4. CONCLUSION
5. TEAM MEMBER CONTRIBUTIONS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *task* | *Bas* | *Mikola* | *Melissa* | *Luca* |
| Coding the sequential model |  |  |  | 4 |
| Coding the rule based model; Difficult cases evaluation |  |  | 5.5 |  |
| Combining all models into 1 cli; Majority baseline; Evaluation | 10 |  |  |  |
| Raw data cleaning;  Logistic regression analysis |  | 5.5 |  |  |
| Working on diagram | 2 | 2 | 2 | 2 |
| Coding and testing the preference prediction |  |  |  | 4 |
| Implementing the dialog management system and transitions | 5 |  |  |  |
| Writing content for the report |  |  | 4 |  |
| Cleanup and coding some transitions; Restaurant preference extractor |  |  |  | 2.5 |
|  |  |  |  |  |
| **Total** | 17 | 7.5 | 11.5 | 12.5 |

Table 1. Contributions

REFERENCES

A  APPENDICES

xxxx

A.1 xxx