ICM : Intent and Conversational Mining from Conversation Logs

Sayantan Mitra, Roshni R. Ramnani, Sumit Ranjan and **Shubhashis Sengupta**
Accenture Labs, Bengaluru
{sayantan.a.mitra, roshni.r.ramnani,}@accenture.com
{sumit.b.ranjan, shubhashis.sengupta}@accenture.com

Abstract

Building conversation agents requires considerable manual effort in creating training data for intents / entities as well as mapping out extensive conversation flows. In this demonstration, we present ICM (Intent and Conversation Mining), a tool which can make the BOT build and update process much faster. ICM can be used to analyze existing conversation logs and help a bot designer to cluster, visualize and analyze customer intents; train custom intent models; and also to map and optimize conversation flows. The tool can be used for first time deployment or subsequent conversational flow updates in chatbots.

1 Introduction

In-spite of the proliferation of GUI based chatbot development environments and availability of open source and commercial tools with low code or no code environments, building chatbots remains a challenge. Many frameworks exist to help a non technical user build chatbots [1] [2], including mechanisms to enter the training data, and drag n drop methods for creating conversation flows. Similarly, research works focus on designing chat bots as end-to-end neural systems or using reinforcement learning based methods. However, limited work exists on techniques to automatically obtain and prepare training data by leveraging existing conversatons that can then be used by commercial tools building task-oriented chatbots.

In this paper , we discuss a tool that takes a user through a guided step by step process of clustering intents, reviewing the intent labels and conversation states, grouping of conversation flows, analyzing individual conversations and, finally exporting the training data for intents and conversation flow. This information can be used by non technical chatbot developers to create chatbots using any of the available chatbot building tools.

The rest of the paper is organized as follows: In section 2 we discuss some of the related work in this space. Section 3 high lights the key features in our tool. Section 4 discusses the technical details of the tool including the algorithms used. 5 discusses the key aspects of the demo. Finally, in section 6 we conclude the paper.

2 Related Work

Multiple methods have been proposed in using realistic data for developing chatbots.Wirén et al. (2007) suggest a modified version of the wizard of the oz approach by collecting transcripts of real conversations between service agents and customers. Many bots are being built to augment human service agents, and hence there is a rich set of information available as human (customer) - human (service agent) conversations.The tool Graph2Bot (Bouraoui et al., 2019) analyzes such existing conversations but fails to create a format that can be leveraged by commercial tools.

In the absence of human conversation logs, text in the form of emails and Service Now tickets can also provide insights about the queries that can possibly be handled by the chatbot. Mallinar et al. (2019) provide a mechanism to bootstrap conversational agents by helping select the necessary training samples.

Once a chatbot is deployed, there are existing tools that perform various forms of conversation analytics. However, these tools do not provide a direct mechanism to leverage these insights back into the enhancement of the chatbot.

3 Key Steps in Analyzing Intent and Conversations

ICM enables multiple people supporting all phases of the chatbot development process to identify the

[1] https://dialogflow.cloud.google.com
[2] https://botmock.com/

Proceedings of the SIGdial 2022 Conference, pages 403–406
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022 ©2022 Association for Computational Linguistics

key intents, the conversation flows, conversation flow analytics including volumetric and temporal analyses, the user sentiment and emotion as well as evolution of conversations over time. This is done in an offline manner by importing the conversation logs. The conversation logs may be human-human conversations captured at the beginning of the chatbot development life cycle, and / or human-bot based conversations at the run phase of the chatbot life cycle. The key features of the tool are as follows:

3.1 Intent Discovery

This is the mechanism by which an automatically extracted short description is used to cluster the conversations in a semi-supervised way. The user is then allowed to select or modify an automatically generated intent label and export the training examples applicable for each intent.

3.2 Intent Analysis

This screen allows the user to view detailed charts on the volumetric analysis (numbers, intensity), temporal (time-of-day, periodicity) characteristics etc. of the intents found.

3.3 Conversation Analysis

The user can view the combined conversation flows per intent or across intents. Through this screen, an analyst can analyze each conversation state, understand the most common flows through the system, identify bottle necks etc.

4 ICM : Technical Details

The tool contains a front end for labelling, analyzing and reviewing existing conversations, as well as the backend containing a rich set of clustering algorithm options, conversation summarization, sentiment and emotion detection options. Figure 1 shows the high level diagram of the tool.

4.1 Front End

The user can upload existing conversation logs or other text data in the form of emails etc via a simple CSV or excel file. The column containing the short description of the content must be identified. The short description, if not present, is generated by using the module described in Subsection **"Conversation Description"**. Privately Identifiable Information is anonymized separately using a custom

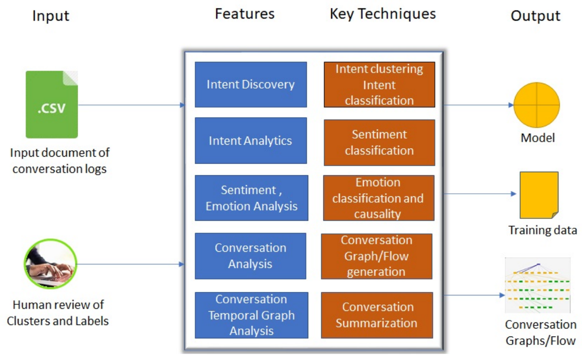

Figure 1: High Level Diagram of ICM

.

python script with regex and spaCy[3] NER model. At the front end, the user can select from a list of clustering algorithms, language models and clustering parameters. The clustered intents, conversation states and flows can be reviewed and labelled in the user interface. The user can export the generated labelled data and also conversation graphs and flows from the front end. Further, the front end also provides interactive visuals for comparison of conversation flows for the users.

4.2 Backend

4.2.1 Conversation Summarization

We use a summarization module based on BART (Lewis et al., 2019) trained on Samsum (Gliwa et al., 2019) data available in the transformers library.

4.2.2 Intent Clustering

The information uploaded into the system goes through three key steps: 1) The user must choose the clustering algorithm algorithm (ITER-DBSCAN (Chatterjee and Sengupta, 2021),HDBSCAN (McInnes et al., 2017)) , sentence embedding (BERT,USE, mBERT), select optional dimentionality reduction (UMAP) and other hyper parameters. The user can run the algorithm multiple times with different configurations and choose the best based on the coverage and the homogeneity of the clusters formed. 2) For each cluster the system provides label suggestions. These are done by using a combination of terms obtained using TD-IDF and the top 5 occurring skip3grams. 3) The system marks similar clusters by calculating the centroid of each cluster and finding the cosine similarity.

[3] `https://github.com/explosion/spaCy`

Figure 2: Screenshot of Sentiment analysis.

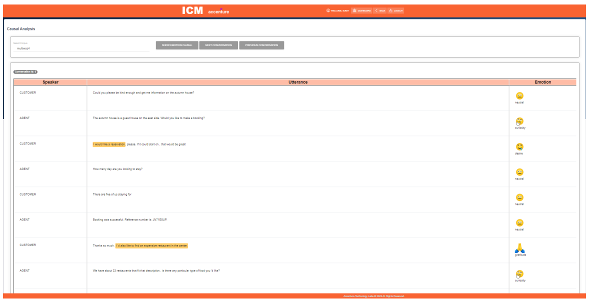

Figure 3: Screenshot of causal analysis.

4.2.3 Sentiment , Emotion and Causality

The tool can identify the user sentiment and emotion per utterance and the causality of each emotion. The sentiment analysis module classifies each utterance into positive, negative and neutral. We use the architecture described by Munikar in (Munikar et al., 2019) trained on the ScenarioSA dataset. The sentiment graph shows the change in sentiment throughout the conversation for Agent and Customer (Fig 2). The emotion analysis module identifies the emotion across 27 categories by using BERT trained on the GOEmotions dataset (Demszky et al., 2020). The causality of each emotion is determined by RECCON (Poria et al., 2020) (Fig. 3).

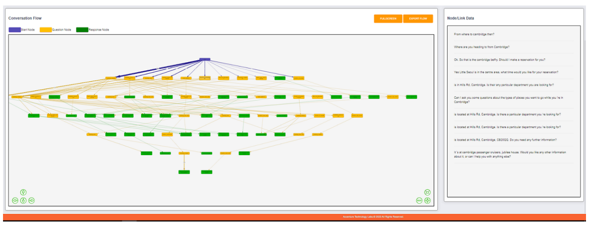

Figure 4: Screenshot of Conversation flow.

Figure 5: Screenshot of temporal conversation graph.

4.3 Conversation Graph and Flow Generation

This module is active only when the uploaded file has the conversation data[4]. Once the file is uploaded into the system, the backend uses a pre-trained CRF based Dialogue Act Classifier (DAC) model to extract the relevant [5] AGENT and CUSTOMER utterances from each conversation. The extracted AGENT utterances are divided into two separate files, viz., questions and responses. Clustering and labelling are done on these two files. After this process, we have labels for each of the relevant AGENT utterances for every conversation.

To generate the conversation graph , we assign a edge between two AGENT state label if there is a transition. For example, the AGENT current state/utterance label is *ques-booking-enquiry* and the next available state is *res-booking-confirm*, so there is a edge between these two former labels and the edge value is the CUSTOMER utterances between these two states. It is a fully connected graph.

To generate a conversation flow (Fig.4), we generated a tree structure. We followed the similar approach as discussed above. The only difference is in the connections. In conversation graph, if the transition is from *ques-booking-enquiry* → *res-booking-confirm* → *ques-booking-enquiry*, we will end up with a loop. But in conversation flow the two *ques-booking-enquiry* are treated separately, that is *ques-booking-enquiry* in level 1 (say) of the tree is different from *ques-booking-enquiry* in level 2 (say). In conversation flow, we also calculate the weight of the edges . For examples, if for all the conversations there is 5 transition between two states then the weight of the edge becomes 5.

The tool can also generate temporal conversation flows (Fig.5) for each intent. This helps the end

[4]Here, we assume the conversation is between AGENT and CUSTOMER.

[5]Extracts only *{QUESTION, COMMAND, INFO}* type utterances and discard other types like *GREETINGS*

Industry Type*	Total Conversations	Identified intents	Note
Telecom1	14000	12	Client wanted to find out the initial conversational flows to increase the containment rate of the conversation. Conversation flow structure generated through ICM are validated by conversational designers of the client.
Telecom2	5000	9	Client shared 5k conversation to determine the intents of the conversation.
Consumer Health	27000	175	Client shared 27k conversation to determine the intents of the conversation.

Table 1: Statistics of ICM tool. *Due to company policy, client names are not disclosed.

user to understand the change in conversation flow either for a single intent over different time period or for different intents over same time period.

5 Demonstration

We will demonstrate ICM on an open source dataset, Multiwoz. During the demo, the audience will see how a user can: a) Upload a conversation dataset, select the appropriate clustering algorithm, language model, and other clustering parameters. b) Label, review and verify the cluster labels and conversation states c) View utterance level details like sentiment, emotion and emotion causality d) View the Conversation Graphs and Trees including the temporal analysis e) View training data and graphs can be exported.

6 Conclusion

In this paper, we highlight a key gap in the existing technology used to build chatbots - the ability to leverage existing data in the form of human-human or human-bot conversations automatically. We discuss a the tool that enables an end user to analyze this data, derive detailed and varied insights and export it in a form that can be leveraged by existing technology to build chatbots.

References

Jean Léon Bouraoui, Sonia Le Meitour, Romain Carbou, Lina M Rojas Barahona, and Vincent Lemaire. 2019. Graph2bots, unsupervised assistance for designing chatbots. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 114–117.

Ajay Chatterjee and Shubhashis Sengupta. 2021. Intent mining from past conversations for conversational agent.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Neil Mallinar, Abhishek Shah, Rajendra Ugrani, Ayush Gupta, Manikandan Gurusankar, Tin Kam Ho, Q Vera Liao, Yunfeng Zhang, Rachel KE Bellamy, Robert Yates, et al. 2019. Bootstrapping conversational agents with weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9528–9533.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu, Romila Ghosh, Niyati Chhaya, Alexander F. Gelbukh, and Rada Mihalcea. 2020. Recognizing emotion cause in conversations. *ArXiv*, abs/2012.11820.

Mats Wirén, Robert Eklund, Fredrik Engberg, and Johan Westermark. 2007. Experiences of an in-service wizard-of-oz data collection for the deployment of a call-routing application. In *Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, NAACL-HLT, Rochester, NY, April 2007.*, pages 56–63. Omnipress.