An Ensemble Model with Ranking for Social Dialogue

Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part
Igor Shalyminov, Xinnuo Xu, Yanchao Yu
Ondřej Dušek, Verena Rieser, Oliver Lemon
The Interaction Lab, Department of Computer Science
Heriot-Watt University, Edinburgh, EH14 4AS, UK
{ac293, v.t.rieser}@hw.ac.uk

Abstract

Open-domain social dialogue is one of the long-standing goals of Artificial Intelligence. This year, the Amazon Alexa Prize challenge was announced for the first time, where real customers get to rate systems developed by leading universities worldwide. The aim of the challenge is to converse "coherently and engagingly with humans on popular topics for 20 minutes".

We describe our Alexa Prize system (called 'Alana') consisting of an ensemble of bots, combining rule-based and machine learning systems, and using a contextual ranking mechanism to choose a system response. The ranker was trained on real user feedback received during the competition, where we address the problem of how to train on the noisy and sparse feedback obtained during the competition.

1 Introduction

This paper discusses two of the major challenges when building open-domain social dialogue systems:

1. How can we facilitate open domain interaction while still executing control?
2. Which utterance fits best in a given dialogue context?

Early systems for social chat, such as ELIZA (Weizenbaum, 1966), were based on carefully handwritten rules, but recent systems are often trained using a variety of (deep) learning techniques over large public data sets, such as OpenSubtitles or Twitter (e.g. Vinyals and Le, 2015; Sordoni et al., 2015; Li et al., 2016). However, learning directly from data also has its pitfalls when deploying a system to real customers, as recent examples such as Microsoft's Tay bot demonstrate. We present a hybrid model, incorporating hand-crafted rules (validated and developed through customer feedback) and machine learning models trained on carefully chosen datasets.

Following previous hybrid systems, (e.g. Yu et al., 2016), we apply a ranker model to select the most relevant reply from a pool of replies generated by an ensemble of different agents/bots. It is still an open question how to best define this ranking function. Previous work has manually defined a evaluation function based on hand-selected turn-level features (Yu et al., 2016; Li et al., 2016). Other work has experimented with learning from crowdsourced user ratings (Lowe et al., 2017). One major drawback of such previous work is that it only evaluates a possible response locally, i.e. per turn, rather than considering its contribution to the overall dialogue outcome, (e.g. to engage the user. As such, these ranking functions often favour safe, but dull responses (Lowe et al., 2017)). We experimented with a variety of ranking functions and datasets as described below. This resulted in one of the top bots in the competition according to average customer rating, as well as with respect to average dialogue length.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

2 System Design and Architecture

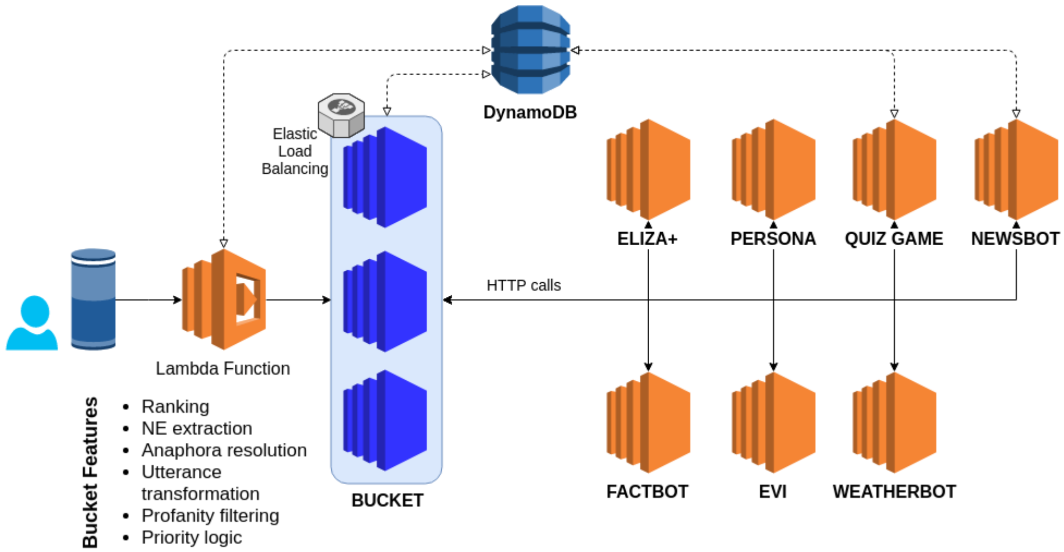

Figure 1: Alana is a hybrid hierarchical architecture with ranking

The system architecture is shown in Fig. 1. We rely on an ensemble of bots. These bots fall into two main categories:

1. **Data-driven Bots:** We experimented with retrieval based bots as well as generative Sequence-to-Sequence models (Seq2Seq, see section 2.1.2) While the former always produce well-formed sentences (as retrieved from the data set), the latter can generate new and possibly more contextually appropriate replies, however at the expense on needing larger data sets to learn from. We follow previous work by combing both paradigms into an ensemble-based approach (Song et al., 2016).
2. **Rule-based bots** are used to respond to the specific user queries in a controlled and consistent way, (e.g. to queries about the personality of our bot, such as favourite things etc., or the weather), using a combination of in-house developed bots and extended versions of 3rd party bots.

These two categories include the following bots:

Persona: A rule-based system implemented in AIML[1] whose main purpose is to maintain personality-based responses consistent across turns, such as music tastes or other preferences. *Persona* also includes replies to other topics, where we want to guarantee an appropriate response to inappropriate user utterances and topics such as sex, as per the competition rules.

Eliza: We extended an existing Eliza-style chatbot called *Rosie*.[2] Since the initial *Rosie* bot was designed for mobile devices, we heavily altered it for the Challenge.

NewsBot: An information retrieval bot based on an open-source framework Lucene.[3] We build and continuously populate a search index of selected news sources provided via NewsAPI.[4] For indexing as well as for the bot's responses, we use summaries of the news articles extracted with an open-source library called *Sumy*.[5] In order to select a relevant piece of news for a user's query, we create 1, 2, and 3-grams over the user's utterance and dialogue context. We employ the BM25 algorithm to score news relevance, with named entities and noun phrases from the user query boosted using a set of weights adjusted empirically. A re-ranking step is then applied for the top 10 candidates based on the articles' recency.

[1] `http://www.alicebot.org/aiml.html`
[2] `https://github.com/pandorabots/rosie`
[3] `https://lucene.apache.org`
[4] `https://newsapi.org`
[5] `https://pypi.python.org/pypi/sumy`

Factbot – Fun facts, Jokes, and Stories: A collection of facts, jokes and stories that get triggered whenever the user specifically asks for them or as a deflection strategy when no suitable response is found. For the fun facts, the user can also specify a named entity ("*Tell me a fact about X*"). Otherwise, a fact is chosen randomly. The data was collected from a multitude of online resources.

Quiz Game: A hand-crafted system developed using a VoiceXML-based structure. During the game, the user has to guess the right answer to topic-specific questions (e.g. 80's music, science, history, sport and geography). The user can end the game at any point.

EVI: A third party bot retrieving factual information (if applicable) about the user utterance, powered by the EVI question answering engine API.[6] This bot returns only one candidate if there is one. Some EVI answers which would not be appropriate in a dialogue are filtered out.

Weatherbot: A simple rule-based bot that provides the user with weather-related information, if asked for, querying the *OpenWeatherMap API*[7] on the fly.

Each of these bots produces a possible system utterance according to its internal rules. Note that not all bots fire at each turn. All the returned candidates are postprocessed and normalized. Profanity, single-word and repetitive (news only) candidates are filtered out. The final system response is selected in three steps:

1. **Bot priority list.** Some of the deployed bots are prioritized, i.e. if they produce a response, it is always selected. The priority order is the following: *Quiz game, Factbot, Weatherbot, Persona, Evi*.
2. **Contextual priority.** The NewsBot's response is prioritized if it stays on the topic of a previously mentioned news story.
3. **Ranking function.** If none of the priority bots produced an answer, the rest of the deployed bots' responses populate the list of candidates and the best response is selected via a ranking function, see Section 4.

In the extreme case where none of the bots produced an answer (or all of them were filtered out due to postprocessing rules), the system returns a random fun fact, produced by the *Factbot*. Please refer to Papaioannou et al. (2017) for more details.

2.1 Other Bots and Data

We also experimented with other data-driven bots, which were not included in the final system.

2.1.1 Data Sets for Information Retrieval Bots

- **OpenSubtitles** (Lison and Tiedemann, 2016), with the automatic turn segmentation provided by Lison and Meena (2016). We used all dialogues of two or more turns and filtered the data as described below.
- **Cornell Movies, Jabberwacky, CNN:** these datasets proved to be too small for our purposes: Cornell Movie Dataset (Danescu-Niculescu-Mizil and Lee, 2011), Jabberwacky chatbot chat logs[8], and CNN chat show transcripts from Yu et al. (2016, 2017).

In order to comply with the competition rules, we first filtered the data for profanities. However, profanities are often context-dependent and hard to capture by a purely lexicon-driven approach. As such, we experimented with restricting the OpenSubtitles data set using age ratings of the movies. We obtained movie ratings from IMDb and only included in our dataset the movies with a U.S. "G" or U.K. "U" ratings ("general", "universal").

Another problem from OpenSubtitles data was the occurrence of many personal names and other named entities that would appear out-of-context in a dialogue. We used Stanford NER (Finkel et al., 2005) to detect named entities and filtered out all context-response pairs containing named entities in the response. However, the downside of this approach is that we ended up with much smaller data sets which made data-driven approaches, such as the generative Seq2Seq approach less feasible.

[6] `https://www.evi.com/)`
[7] `https://openweathermap.org/`
[8] `http://www.jabberwacky.com/`

2.1.2 Seq2Seq

Throughout system development, we experimented with a sequence-to-sequence dialogue model (Vinyals and Le, 2015), training it on several datasets. The first promising behaviour was obtained with Twitter data[9]: it was interesting and mostly grammatical yet often offensive and politically related. We then switched to a subset of Reddit logs – over 21,000 conversation snippets in the form of question-answer pairs cleaned from profanity and filtered to only contain small-talk conversation (thanks to Dr. Zhuoran Wang). In order to exclude ungrammatical responses, we disregarded all answers with a low confidence score (defined as the sum of the logits at the decoder's output). We adjusted the confidence threshold empirically on a separate development set of 100 sample user utterances both collected from WoChat[10] transcripts and paraphrased from a list of popular daily topics provided by Amazon.

The experiment thus resulted in a casual conversation bot: its answers are supposed to be given at times when the user is following up on the previous system's answer or just hesitating. Due to time constraints, the final version of the seq2seq bot was not deployed into production, and so its possible contribution to the users' ratings is left for future work.

3 Example Dialogue

Note: The dialogue presented here does not come from real customer data, but was recreated by interacting with our system (running a text-based version on Telegram) The same structure of interaction and Named Entities are used as occurred in a real dialogue. We also indicate which bot in our ensemble generated the answer.

A: Hi, [...] What would you like to talk about?
U: music
A: *(Persona)* Great! Who is your favorite singer?
U: Bob Dylan
A: *(Evi + additional question)* Bob Dylan is an American songwriter, singer, painter, and writer. What are your opinions on Bob Dylan?
U: When was **he** born?
A: *(Evi + anaphora resolution)* Bob Dylan was born on Saturday May 24th 1941.
U: What's happening with **him**?
A: *(News-multi-turn + anaphora resolution)* I heard this on the radio – Was Dylan too freewheelin' in borrowing for Nobel lecture? The whiff of plagiarism is blowin' in the wind for Bob Dylan. Want to know more?
U: sure
A: *(News-multi-turn)* It seems that you are interested in this topic. I'm so glad. Here's more. Here's Dylan: Some men who receive injuries are led to God, others are led to bitterness The cases Pitzer found are not blatant or explicit – there are no verbatim sentences, only identical phrases and similar phrasing. What are your opinions on Bob Dylan?
U: I love him!
A: *(Persona)* Great! What else do you like?
U: cooking
A: *(Facts)* I heard that every day, Americans eat an estimated 18 acres of pizza.

4 Experiments with Ranking Functions

The responses proposed by each bot are ranked according to a set of features. We have experimented with several ranking functions.

4.1 Hand-engineered Ranker function

The hand-engineered ranking function uses the following features:

[9] `https://github.com/Marsan-Ma/chat_corpus`
[10] `http://workshop.colips.org/wochat/`

- **Coherence:** Following Li et al. (2016), we reward semantic similarity between the user's utterance and the candidates using Word2Vec (Mikolov et al., 2013)
- **Flow:** Also similar to Li et al. (2016), we penalise similarity between consecutive system utterances in order to prevent repetition. Here, we use both Word2Vec and METEOR word n-gram overlap as measures of similarity.
- **Questions:** By promoting questions, we aim to incite the user to continue the conversation.
- **Named Entities:** We strongly reward utterances containing the same named entities as the user's reply to promote candidates relating to the same topic.
- **Noun Phrases:** Similarly, we reward matching noun phrases between the user's and the system's utterances. Noun phrases are identified based on part-of-speech tagging.
- **Dullness:** We compare each response to a list of dull responses such as "I don't know" and penalise Word2Vec similarity between them, since we would like the bot's utterances to be engaging, similarly to Li et al. (2016).
- **Topic Divergence:** We trained a Latent Dirichlet Allocation (LDA) model on a weighted combination of preprocessed versions of the OpenSubtitles and the WashingtonPost datasets. We set the vocabulary size to $20k$ and the number of topics to 200, and we used a tailored stop-words list. For every proposed answer in the bucket, we compute the topic divergence from the user utterance.
- **Sentiment Polarity:** We use the VADER sentiment analyser (Gilbert and Hutto, 2014) from the NLTK toolkit,[11] which provides a floating point value indicating sentence sentiment.

These features are calculated using the last two system turns in order to maintain dialogue context. The final score is a weighted sum of these features:

$$\begin{aligned} score = 0.25 * turn_0 + 0.25 * turn_1 + 0.25 * turn_2 + 0.25 * noun\_phrases \\ +3 * named\_entities - 0.25 * topic\_divergence \end{aligned} \tag{1}$$

where $turn_i$ is computed using the i-th utterance counting from the end of the dialogue history:

$$\begin{aligned} turn_i = -0.2 * flow_{sem\_similarity} - 3 * flow_{METEOR} + 0.1 * coherence_{sem\_similarity} \\ -0.24 * dullness + 0.2 * question + 0.1 * sentiment\_polarity \end{aligned} \tag{2}$$

4.2 Linear Classifier Ranker

In order to use the feedback ratings obtained from real users in the competition, we also trained the VowpalWabbit linear classifier (Langford et al., 2007) to rank Bucket responses based on the following features:

- bag-of-n-grams from the context (preceding 3 utterances) and the response (unigrams, bigrams, and trigrams)
- position-specific n-grams at the beginning of the context and the response (first 5 positions)
- dialogue flow features, same as for the hand-engineered ranker (see Section 4.1)
- bot name.

The ranker is trained as a binary classifier, but it outputs a floating-point score in practice. At runtime, the highest-scoring response is selected for the output.

We initially trained the ranker on Cornell movies, Twitter, and Jabberwacky datasets (see Section 2.1.1), with positive examples from the real dialogues and negative ones randomly sampled from the rest of the set, but the ranker only learned to prefer responses similar to data from these datasets; its performance in real dialogues was lacking in our tests. Therefore, after collecting enough live dialogues during the Alexa Prize competition, we retrained the ranker on *real dialogues collected during the competition*. The rating target function is an approximation of human ratings – we use all context-response pairs from successful dialogues (human rating 4 or 5) as positive examples (value +1) and all pairs from unsuccessful dialogues (rating 1 or 2) as negative (value -1) and train the ranker to mimic this rating.

[11] `http://www.nltk.org/api/nltk.sentiment.html`

We collected 60k dialogue instances over one month for training and 7k dialogue instances over 4 days as a development set. We did not perform any large-scale parameter optimization, but based on performance on the development data, we selected the following VowpalWabbit parameters:

- logistic loss function (logistic regression),
- feature concatenations (context + response n-grams, pairs of n-grams from responses, bot name + response n-grams, bot name + context n-grams, bot name + dialogue flow, bot name + context n-grams + response n-grams),
- 16-bit feature hash table,
- 1 pass over the training data.

This setup reached 69.40% accuracy in classifying the development data items as positive or negative. The results of deploying this Linear Ranker are presented in section 4.3.

4.3 Results

The Linear Ranker, trained on the user feedback received during the competition (see Section 4.2), was deployed on top of Alana v1.1, and evaluated in comparison to the hand-crafted ranking function (see Section 4.1). The results are shown in Table 1.

System	average user rating	number of dialogues
Alana v1.1 : Hand-engineered Ranker	3.26	191
Alana v1.1 : Trained Linear Ranker	3.28	272

Table 1: Results: Trained Linear Ranker (semi-finals period)

This shows that we can continuously improve system performance by training on real customer feedback from the competition, even though it is noisy and sparse (ratings are only available for whole dialogues, and not each dialogue turn).

5 Future Work

This paper describes our Alexa system as entered in the semi-finals (July-August 2017). We are now competing as one of three systems in the Amazon Alexa Challenge finals, where we have replaced the linear ranker with a neural model. This neural ranker is trained on an increased number of user ratings, which we were able to gather August-October 2017, and outperforms the linear ranker in terms of accuracy.

Acknowledgements

We would like to thank Helen Hastie and Arash Eshghi for their helpful comments and discussions.

References

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. In *Proc. CMCL*, pages 76–87.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. ACL*, pages 363–370.

Gilbert, C. J. and Hutto, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *AAAI Conference on Weblogs and Social Media*, pages 216–225, Ann Arbor, MI, USA.

Langford, J., Li, L., and Strehl, A. (2007). Vowpal wabbit online learning project.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. (2016). Deep Reinforcement Learning for Dialogue Generation. In *Proc. EMNLP*.

Lison, P. and Meena, R. (2016). Automatic Turn Segmentation for Movie & TV Subtitles. In *2016 IEEE Workshop on Spoken Language Technology*.

Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proc. LREC*, Portorož, Slovenia.

Lowe, R., Noseworthy, M., Serban, I., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017). Towards an automatic turing test: Learning to evaluate dialogue responses. In *ACL*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Papaioannou, I., Cercas Curry, A., Part, J. L., Shalyminov, I., Xu, X., Yu, Y., Dušek, O., Rieser, V., and Lemon, O. (2017). Alana: Social dialogue using an ensemble model and a ranker trained on user feedback. In *Proc. AWS re:INVENT*.

Song, Y., Yan, R., Li, X., Zhao, D., and Zhang, M. (2016). Two are better than one: An ensemble of retrieval- and generation-based dialog systems. *CoRR*, abs/1610.07149.

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. In *Proc. NAACL-HLT*.

Vinyals, O. and Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Weizenbaum, J. (1966). ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM*, 9(1):36–35.

Yu, Z., Black, A. W., and Rudnicky, A. I. (2017). Learning Conversational Systems that Interleave Task and Non-Task Content. In *Proc. IJCAI*, Melbourne, Australia. arXiv:1703.00099.

Yu, Z., Xu, Z., Black, A. W., and Rudnicky, A. I. (2016). Strategy and Policy Learning for Non-Task-Oriented Conversational Systems. In *Proc. SIGDIAL*, Los Angeles, CA, USA.