NIPS Conversational Intelligence Challenge 2017 Winner System: Skill-based Conversational Agent with Supervised Dialog Manager

Idris Yusupov [1] and **Yurii Kuratov** [1, 2]
[1] Department of Computational Linguistics,
Moscow Institute of Physics and Technology

[2] Neural Systems and Deep Learning Lab,
Moscow Institute of Physics and Technology
{i.yusupov, yurii.kuratov}@phystech.edu

Abstract

We present *bot#1337*: a dialog system developed for the 1st NIPS Conversational Intelligence Challenge 2017 (ConvAI). The aim of the competition was to implement a bot capable of conversing with humans based on a given passage of text. To enable conversation, we implemented a set of skills for our bot, including chit-chat, topic detection, text summarization, question answering and question generation. The system has been trained in a supervised setting using a dialogue manager to select an appropriate skill for generating a response. The latter allows a developer to focus on the skill implementation rather than the finite state machine based dialog manager. The proposed system bot#1337 won the competition with an average dialogue quality score of 2.78 out of 5 given by human evaluators. Source code and trained models for the bot#1337 are available on GitHub.

1 Introduction

A conversational or a dialogue agent is a system that interacts with a human via voice or text messages. Dialog systems can be task-oriented, i.e. supposed to solve specific tasks such as reserving flight tickets, or general purpose. Another way to differentiate dialog systems is to compare them by domain: closed or open one. When open-domain dialog systems cover a wide range of supported topics, closed-domain are usually specialized on a few topics.

The early dialog systems used a rule-based approach to control dialog flow (Weizenbaum, 1966; Wallace, 2009). Rule-based systems are hard to maintain and new rules should be created for each new domain. This approach is still widely used in cases where the full control of dialog is crucial (e.g., production systems). The availability of dialog datasets (DSTC (Williams et al., 2016), Twitter Corpus (Ritter et al., 2010), Ubuntu Dialogue Corpus (Lowe et al., 2015)) makes it possible to train end-to-end dialog systems (Sordoni et al., 2015; Vinyals and Le, 2015). End-to-end systems are usually based on recurrent neural networks (Graves, 2013) and sequence-to-sequence models (Sutskever et al., 2014), and use raw dialogues as a training data.

One of the main challenges in the field of open-domain dialogue systems is their evaluation. The reason for that is the absence of good evaluation metrics. While goal-oriented systems can be evaluated with the percentage of dialogues when user's task was accomplished, there is no formal measure of the dialogue quality for open-domain systems. The goal of such systems is to generate responses which suits the context. The quality of the response can be measured by its perplexity measure (Serban et al., 2015) which, however, can not assess the adequacy of an answer. In order to measure it researchers often use metrics that compare a generated string to some oracle answer, e.g. BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007) originally used for evaluating Machine Translation model. This is not an optimal way of evaluating chatbots either, because a relevant answer can be different from an oracle (Liu et al., 2016). In recent works on the evaluation of dialogue systems the authors suggest training evaluation metrics on a set of human-labelled dialogues (Lowe et al., 2016; Lowe et al., 2017).

Proceedings of the 27th International Conference on Computational Linguistics, pages 3681–3692
Santa Fe, New Mexico, USA, August 20-26, 2018.

However, obtaining such corpora takes much time and effort, so those metrics are rarely used up to our knowledge.

A competition of open-domain dialogue systems is one way to organize a large-scale evaluation by humans. There are some competitions in the field such as Loebner Prize annual competition (Mauldin, 1994) and Alexa Prize Competition (Ram et al., 2018). The First Conversational Intelligence Challenge was organized as a part of NIPS conference in 2017. The task was to build a conversational agent, which discusses a given text with a human.

A typical competition bot is equipped with a set of skills, such as greeting, asking a question on a certain topic, etc., and decides when to use a specific one. The main challenge here is to understand when to use a specific skill[1], in other words, to develop a dialog manager (DM). The finite state machine (FSM) can be seen as an instrument of the first choice in rule-based dialog managers. The main advantage of the FSM is a deterministic and transparent control of the conversation, but its complexity quickly grows for large systems. Updating of large FSM with new rules is usually hard due to the high volume of required conflict resolutions between parts of the system. This results in the lack of scalability for the FSM approach.

2017 Loebner Prize winner bot Mitsuku[2] uses a rule-based approach based on Artificial Intelligence Markup Language (AIML) (Wallace, 2009). 2017 Alexa Prize participant MILABOT conversational agent (Serban et al., 2017) uses a combination of deep learning, reinforcement learning and rules. MILABOT agent team used a lot of human-annotated data collected from Amazon Mechanical Turk to train the dialog manager. Mitsuku requires a lot of rules, so it is hardly scalable, while MILABOT requires a lot of human-annotated data. In addition those bots do not initiate a dialog with the user, but only give replies to a user.

Instead, we implemented a supervised DM for our bot. It takes the dialog context[3] as an input and outputs the label of the skill to be used. This simplifies developing the proposed DM in comparison with the FSM and allows to focus on the skill development.

Next sections describe our conversational agent *bot#1337* that won the NIPS Conversational Intelligence Challenge. Section 2 describes the implemented skills and Section 3 describes Dialogue Manager (DM). Section 4 gives a high-level view on how skills, DM and user interact together. Section 5 provides detailed information about the participation in the NIPS Conversational Intelligence Challenge, including comparison with other winning system, analysis of skills usage frequency, dialog manager performance and dialogs reading insights. Information about the bot#1337 license, source code and demonstration given in Section 7. To show how the bot works in practice, we have included a sample dialog between the bot and user in the Appendix A.

2 Skills

Discussion of the text require different skills. First of all, a bot should be able to greet the user, to answer and to ask questions about the text. Secondly, text summarization skill may be required, because the user may not read a long paragraph. Often one just wants to chit-chat with a bot without a thorough discussion.

Chit-chat skills Chit-chat skills are required to discuss some common topics, which are often not connected with a given text. We have built three following skills for the chit-chat.

OpenSubtitles and Facebook news seq2seq chit-chat skills use a neural machine translation attention-based sequence-to-sequence models. They are trained using OpenNMT framework. OpenNMT is a generic deep learning framework mainly specialized in sequence-to-sequence models covering a variety of tasks such as machine translation, summarization, image to text, and speech recognition. The framework has also been extended for other non sequence-to-sequence tasks like language modelling and sequence tagging (Klein et al., 2017).

[1] We defined *skill* as a model that takes text and dialog context as an input and outputs a response.

[2] `http://www.mitsuku.com`

[3] We defined *dialog context* as the concatenation of dialog history and the current user utterance.

Training performed with word-level representations and ADAM optimizer. Word embeddings dimension was set to 150 and their weights are updated during the training.

OpenSubtitles seq2seq skill uses the dialog context as an input and generates an utterance as an output. The model is a 2-layer encoder-decoder LSTM with 2048 hidden units. It is trained on the OpenSubtitles dataset (Tiedemann, 2009).

The Facebook news seq2seq skill, in addition to the dialog context, may use a paragraph as an input. Paragraph and dialog context divided with special ”EOP” (end of paragraph) token. It also generates an utterance as an output. The model is a one-layer encoder-decoder LSTM with 1024 hidden units. It is trained on the Facebook news dataset[4], which consists of posts and comments from Facebook.

The Alice (Wallace, 2009) *chit-chat skill* uses the dialog context as an input and generates an utterance as an output. It is built upon AIML rules. We used the open source version of Alice from GitHub[5].

Sequence-to-sequence chit-chat skills generate several response candidates. Often generative models may generate short responses or responses with many identical or undesirable words. These candidate responses are processed by a filtering algorithm (Alg. 1) and if there is no candidate responses left after filtering, then the Alice skill is executed or one of the escape plan template responses is selected. Escape plan responses are a set of pre-defined utterances, such as “Do you like this text?”, “What do you think about the competition?”.

Algorithm 1: Filtering candidate responses

Data: candidate responses
Result: filtered responses
remove duplicate responses;
remove short responses;
remove responses with majority of identical words;
remove responses with majority of stopwords;
remove responses with undesirable words;
return filtered responses;

Q&A skills Discussion of text often consists of asking and answering questions. We developed three skills to generate questions and to answer the questions about the text.

For the question-asking skill we reproduced the feature-rich sequence-to-sequence with attention model (Zhou et al., 2017). The model is a 2-layer encoder-decoder GRU with 512 hidden units. It is trained on the SQuAD dataset (Rajpurkar et al., 2016) using word-level representations and the OpenNMT framework with ADAM optimizer. Word embeddings dimension was set to 300. Lexical and answer features embedded to 32-dimensional vectors. This skill takes a paragraph, named entities and lexical features extracted using Stanford CoreNLP (Manning et al., 2014) as an input and outputs a question. In the training phase, all the SQuAD answers were used to generate questions. During the inference the model used named entities extracted from the text by the Stanford CoreNLP. We use these named entities as an answers and to generate questions.

The answer-checking skill is connected with the question-asking skill. Right answer on each question is prepared by question-asking skill described above. This skill checks the user’s answer using fuzzy matching algorithms (e.g. based on Levenshtein distance). Afterwards, it uses a template to generate a response to the user, such as “You can do better! Hint: first 3 letters is *goo*”.

The question-answering skill uses the bidirectional attention flow (BiDAF) model (Seo et al., 2016). It takes a user’s question and the relevant text as an input and outputs an answer. Finally, these answer were added to template phrases, such as “I think the answer is ...”. We used the open source version of BiDAF model from GitHub[6].

[4]https://github.com/jbencina/facebook-news
[5]https://github.com/sld/convai-bot-1337/tree/master/ALICEChatAPI
[6]https://github.com/allenai/bi-att-flow

Summarization skill Usually, a text passage to discuss is a long (5-10 sentences) read. Text summarization skill helps to save time and to engage the user.

We used a sequence-to-sequence model with attention pretrained on a Gigaword dataset with OpenNMT[7]. The model is a 2-layer encoder-decoder LSTM with 500 hidden units. Word embeddings dimension was set to 500. We applied the model to chunks of a provided text to generate possible summaries. Finally, these summaries were added to template phrases, such as “Maybe this article’s main idea is ...”. To select the best response, we used Alg. 1.

Topic detection skill Mentioning the main topic of the text may engage the user in a conversation. We analyzed data from the first round of the NIPS Conversational Intelligence Challenge and it proved our hypothesis.

The topic detection skill is built upon the BigARTM topic modeling framework. BigARTM is a tool to infer topic models, based on a technique called Additive Regularization of Topic Models. This technique effectively builds multi-objective models by adding the weighted sums of regularizers to the optimization criterion (Vorontsov et al., 2015).

The model takes text as an input and outputs its possible topics. Then, detected topic names are added to template phrases, such as “Am I right that topic of the text is ... ?”. We used the Wikipedia corpora from the BigARTM website datasets sections[8]. The model was trained[9] to predict 15 topics. Topic names such as ”Politics”, ”Culture”, etc., were set up manually by analyzing top-tokens.

Additional skills We implemented two skills that are independent from the input. They both have handwritten sets of phrases, which are randomly selected to be sent to the user.

The greeting skill is used in the beginning of the conversation if the user does not say anything after some time. It includes such phrases as ”Well hello there!”, ”Hiya!”, etc.

The common phrases skill is used when the user does not say anything for some time or when the used skill output is empty. It includes such phrases as ”What do you think about ConvAI competition?”, ”Do you like this text?”, etc.

3 Dialog manager

The DM (Fig. 1) runs two classifiers on user utterance (and, possibly, dialog context) to detect what skill should produce an answer.

The first classifier works on a small supervised set (few key phrases for each skill) and is based on the mean word embeddings (we used GloVe embeddings[10]) and the k-nearest neighbors classifier (k-NN).

The second classifier is based on a large training corpora and we used the fastText library for this case. FastText is a library for efficient text classification and representation learning (Joulin et al., 2016). The fastText supervised model was trained[11] to predict 5 classes (Fig. 1).

All implemented skills can be one of two types: with or without a training dialog dataset. For chit-chat and question-answering skills we have datasets with utterances, questions, and answers. It allows us to train the classifier to select which dataset (and it infers which skill) is more suitable for the incoming user utterance. Summarization, topic detection, and the ask-question skills do not have good common known dialog datasets, which provide utterances that should activate these skills. For these skills we wrote few key phrases (3-10 for each skill) by our own and used them with k-NN classifier based on mean word embeddings.

For example, we have 2 skills - Open Subtitles and Topic Detection. First skill classifier is trained with fastText, because it has many dialog utterances (e.g. ”They still behind us?”, ”Senora, give me a break!” and many others). Second one does not have any dialog utterances, so we should add a few of

[7] `http://opennmt.net/Models/#english-summarization`

[8] `http://docs.bigartm.org/en/stable/tutorials/datasets.html`

[9] Training script available at `https://github.com/sld/convai-bot-1337/blob/master/topic-modelling`

[10] `https://nlp.stanford.edu/projects/glove/`

[11] Training script available at `https://github.com/sld/convai-bot-1337/tree/master/classifiers/factoid_question_vs_all`

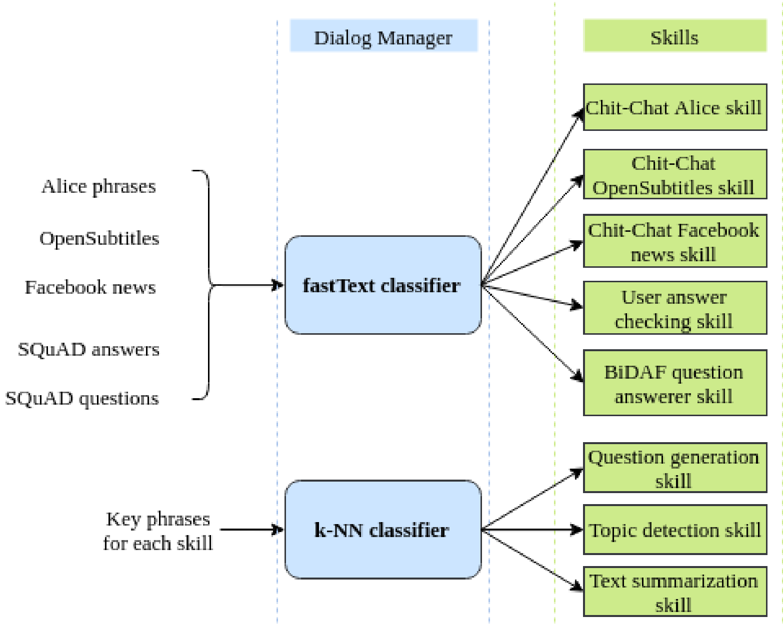

Figure 1: Dialog manager: classifiers, datasets and skills corresponding to them.

them ourselves to train a k-NN classifier (e.g. ”What is the theme of text?”, ”Say me theme”, ”This text main topics”).

Skills without dialog datasets have a higher priority; if the k-NN classifier prediction confidence level exceeds the threshold value (90%), then the skill selected by this classifier is used to generate the response. Otherwise, the skill is selected by the fastText classifier.

4 Dialog system and user interaction flow

Dialog system and user interaction flow (Fig. 2) shows how user interacts with a dialog system.

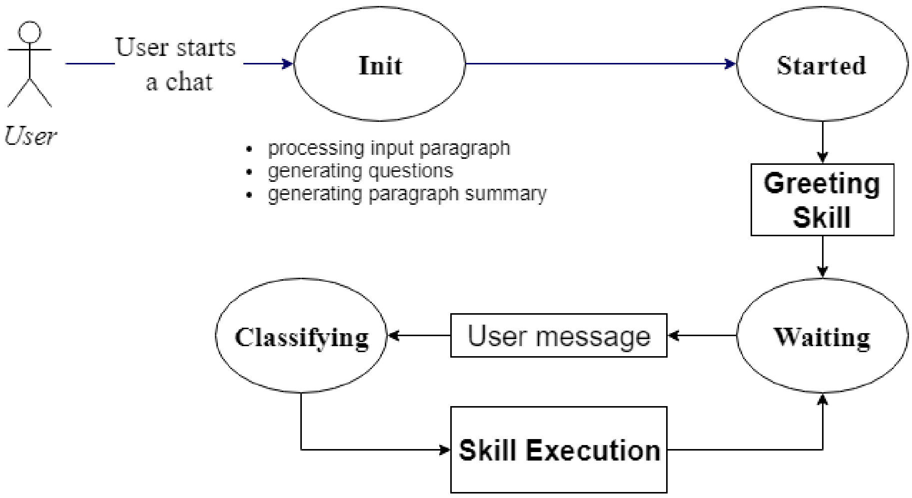

Figure 2: Dialog system and user interaction flow.

The dialog agent initialization begins when a user starts a chat. At the **init state**, the DM and skills are initialized. Some skills, such as question generation and text summarization, process input paragraphs and generate responses for further usage. Thus, the DM can choose from these responses to send a message to the user.

After that, the dialog agent switches to the **started** state. A user or a bot can start the conversation.

Thus, at this state, the bot waits for the first user message or executes the greeting skill if there is no message from the user after some time.

Answering to user message is the most important task to handle. When receiving a message from the user, the **classifying** state activates. At this state, the DM decides which skill to use. The DM runs two classifiers on user utterance (and, possibly, dialog context) to decide which skill should produce an answer. After that, the bot executes the skill and sends a generated message to the user.

After the skill execution, the bot goes to the **waiting** state. This state is about engaging the user in conversation when he or she is not talking to the bot. If after some time there is no message from the user, the bot tries to motivate the user to talk. It executes a random skill, such as question asking, and sends a message to the user. If there is no reaction from the user for a long time, then the bot ends the conversation.

5 Participation in the Conversational Intelligence Challenge

The main task of the NIPS Conversational Intelligence Challenge 2017 (ConvAI[12]) was to build a conversational agent, which discusses a given text with a human. Each dialog was evaluated by a human at the end of the conversation on a scale from 1 (bad) to 5 (excellent) in three dimensions: quality, breadth, and engagement. Each bot utterance could be also marked as appropriate or not. The first round dataset (2778 dialogues) was released at the end of the first human evaluation round[13].

The early version of our bot had a chit-chat, Q&A skills and a dialog manager based on rules and fastText classifier. This approach took the second place in the first round of the Conversational Intelligence Challenge with an average quality score 2.31 out of 5. A winner bot took 2.38 out of 5, while human 3.8 out of 5. The dataset released after this round provided insights on how to improve our conversational agent and helped to detect undesirable behavior. Particularly we decided to add a chit-chat Alice, paragraph's topic mention to the greeting and text summarization skills. Newly added skills required new logic to support. We enhanced the dialog manager by adding k-NN classifier for skills without dialog data described in Section 3. It helped us to avoid using rules and simplified conversational agent's development. As a result of introducing the above-mentioned changes, our bot won the Conversational Intelligence Challenge Finals (Table 1).

Bot name	Quality (average human evaluation score)	# dialogues
bot#1337	2.779	68
poetwannabe	2.623	53
RLLChatBot	2.462	13
kAIb	2.2	35
PolyU	1.75	28
DeepTalkHawk	1.262	42

Table 1: Results of the Conversational Intelligence Challenge Finals.

5.1 Comparison with *poetwannabe*

Poetwannabe is the other winning system from University of Wroclaw which also used a multi-skill architecture, but focused on extracting knowledge from external resources (Wikipedia, DBPedia, Wikiqoute). Instead of supervised dialog manager to control dialog flow they used confidence scores from each skill, based on linguistic features, dialog history and internal state (Chorowski et al., 2018).

5.2 ConvAI finals dataset analysis

To understand better bot's issues and to develop further its improvement strategy, we analyzed the frequencies of skills usage, dialog manager performance and user dialogues of the Conversational Intelligence Challenge finals dataset. The dataset was provided by the competition organizers. We have

[12]http://convai.io/2017/
[13]http://convai.io/2017/data/

manually labeled 68 dialogs with our bot. Each bot's utterance in the dialog was tagged with predicted skill and true skill labels. Predicted skill label is the one selected by the supervised dialog manager. True skill label is the one that we found to be the most appropriate for the dialog context.

5.2.1 Frequency of skills usage

Frequency of skills usage was calculated by predicted skill label grouped by a dialog score for each dialog (Table 2). After analyzing the frequency of skills usage we saw that chit-chat skills should be carefully implemented. Chit-chat skills were most frequently selected by the dialogue manager. Skills like topic detection or summarization are used rarely because usually they are executed only one time during the dialog. It does not imply that summarization or topic detection skills are unnecessary. They provide good user experience in a conversation. A fragment of best-scored dialog shows a positive user reaction to a summarization skill:

User: What do you think about this text?

Bot: I suppose that this text main idea is: New rules on investment?

User: Maybe you are right

Other fragment with worst-scored dialog shows negative reaction for chit-chat skills:

Bot: What country borders the south of guinea-bissau?

User: I don't know. Can you tell me?

Bot: I don't know.

User: Senegal is to the north, isn't it?

Bot: I will not.

Dialog score	Chit-chat skills, %	Q&A skills, %	Summarization, %	Topic Detection, %	Greeting, %	Common Phrases, %
5	71.4	16.3	0.7	3.2	3.9	4.6
4	56.2	23.3	2.7	6.8	8.2	2.7
3	57	21	1.1	5.9	7.5	7.5
2	55.3	24.9	2.3	6.8	7.6	3
1	64.3	15.6	1.7	4.5	6.7	7.2

Table 2: Bot#1337's percentage of skills usage in the ConvAI Finals dialogs. They are grouped by a dialog score (human evaluation score). We count how many times each type of skill was invoked in each dialog. Chit-chat skill is the one of OpenSubtitles, Facebook news and Alice skills. Q&A is the one of question-asking, answer-checking and question-answering skills.

5.2.2 Dialog manager performance

By using true skill and predicted skill labels, F1 measure was calculated. Topic detection, greeting, common phrases skills were not included to the calculations because they were executed in a waiting state almost every time (described in Section 4) and dialog manager was not used here.

Skill Name	Precision	Recall	F1	Total utterances count
OpenSubtitles chit-chat	0.79	0.84	0.81	249
Answer-checking	0.82	0.70	0.75	57
Alice chit-chat	0.81	0.68	0.74	152
Facebook chit-chat	0.63	0.85	0.73	53
Question-asking	1.00	0.56	0.71	9
Summarization	0.67	0.57	0.62	14
Question-answering	0.56	0.60	0.58	25
Average	*0.77*	*0.76*	*0.76*	*559*

Table 3: Dialog manager performance

Bot#1337's supervised dialog manager (described in Section 3) average F1 equals to 0.76. Question-asking and summarization skills that were selected by the k-NN classifier show lower F1 score than skills selected by the fastText classifier (chit-chat and answer-checking skills). F1 for these skills can be improved by adding more examples to the k-NN classifier. However this approach does not apply to a question-answering skill. It is selected by the fastText classifier and has the lowest F1 in the Table 3. The error analysis showed that the user often asks complex questions that are not covered by the training set from SQuAD. For example:

User: Let's talk about Guinea-Bissau then. What do you know about it?" [classified as Open Subtitles]

But even with the perfect F1 score, analysis of how each skill works in a dialog is necessary. Dialog reading helps to analyze skills errors, to find new skills that should be developed and to find the direction of bot's improvement.

5.2.3 Dialogs reading insights

After reading 68 dialogs with our bot from the finals, we can conclude that having diverse and high quality chit-chat skills and engaging a user into talking when he or she is not responding for some time (waiting state described in the Section 4) helped the bot to make a conversation better.

Other insights that will be able to help to improve our bot in the future described below.

The bot starts to talk fast and a user complains to it.

Bot: Yo! [Greeting skill]

Bot: I think this text is connected with nationalities. I can ask you a question about it. [Topic detection skill]

User: Wait, I am still reading it

Bot: Why is this news? [OpenSubtitles chit-chat skill]

It would be better to have a separate skill to detect if a user still reading the text and reply to the user with "Okay, I am waiting" message.

A user asks the bot about the competition, but it does not have the skill to response.

User: What are my instructions for today?

Bot: I'm not sure. [OpenSubtitles chit-chat skill]

User: Do you something about CONVAI?

Bot: No, i'm not. [OpenSubtitles chit-chat skill]

Lack of semantic model of the text does not allow the bot to talk about the facts in the text.

User: What kind of sport is this text about?

Bot: I suppose that this article can be described as: Win club world cup..? [Text Summarization skill]

User: But what sport? Football, hockey, basketball?

Bot: I'd like to say that correct answer is: barcelona accomplished two new records in spanish [Question-answering skill]

User: I want to know more about Barcelona

Bot: Tell me about your likes and dislikes? [Alice Chit-chat skill]

User: Ok, is this article about football?

Bot: I think this article main idea is: Season begins. [Text Summarization skill]

We have the Q&A skills, but they do not cover cases like in the dialog above. A high quality conversation about the facts in text requires using knowledge bases like DBpedia. Besides it is worth to use some Web APIs (e.g. BabelNet knowledge base at `https://babelnet.org`), but rules of the ConvAI competition do not allow to do it. According to the rules everything should work without any connection to external services.

Sometimes the bot is not polite.

User: It was basically my answer
Bot: February was basically his answer? very interesting. [Alice Chit-chat skill]
User: What did you base my gender on?
Bot: Nothing. [OpenSubtitles chit-chat skill]
User: Then why ”his answer”? And why talk to me in the third person?
Bot: Oh darn, I broke you. [OpenSubtitles chit-chat skill]

Answer-checking skill and question-answering skill should be more intelligent. Answer-checking skill sees the difference between a number and a word representing that number. But it should not and dialog below describes the case:

Bot: How many mathematical treatises have managed to survive? [Question-Asking skill]
User: 3?
Bot: It is incorrect:) Hint: first 3 letters is thr. Try again [Answer-checking skill]
User: 3
Bot: No [Answer-checking skill]
Bot: I think that answer is: three [Answer-checking skill]
User: Infact I did mentioned the same
Bot: How old are you? [OpenSubtitles chit-chat skill]

Sometimes users ask the bot to answer the question asked by the bot in the first place. However, the bot can’t answer if the user uses anaphora constructions.

User: Ask me question
Bot: How many years did nbc universal sell in 2006? [Question-Asking skill]
User: I dont know
Bot: Why wouldn’t they have taken the meeting? [OpenSubtitles chit-chat skill]
User: Do you know?
Bot: I dont know what you’re doing. [OpenSubtitles chit-chat skill]

User tries to teach the bot, but the bot does not understand it.

User: What was article about?
Bot: Maybe this text can be described as: Father and grandfather born in hungary..? [Text Summarization skill]
User: No! It was about von Neumann
User: Which city lies on seven hills?
Bot: What a joke. [OpenSubtitles chit-chat skill]
User: Answer me
Bot: Try asking me again using simpler words. [Alice chit-chat skill]
User: Which city lies on seven hills. Answer: Rome.
Bot: Why is this news? [Facebook news chit-chat skill]

Bot’s conversation style is the same. It may be worth to make different conversation styles to engage the users who have already talked to a bot.

6 Conclusion

In this paper, we have described the conversational agent *bot#1337*. This bot took the first place in the NIPS Conversational Intelligence Challenge 2017 with an average dialogue quality score of 2.78 out of 5. The supervised dialog manager allowed us to focus on the skills implementation, to avoid rules and to simplify the development of a bot. Skills like topic detection, Q&A, text summarization are engaging a user into a conversation about the given text. According to the conversational data received during the competition, a user often chit-chats with a bot. AIML-based Alice chit-chat skill covers a wide range of topics, while sequence-to-sequence based chit-chat skill trained on Open Subtitles dataset makes a conversation more enjoyable. Open source machine learning software like OpenNMT, BigARTM, fastText gives us a big boost on the conversational agent's implementation. By making the framework for *bot#1337* publicly available, we gave an opportunity for the community to get acquainted with its various modules and to develop further the conversational intelligence.

7 Availability

Conversational agent demonstration is accessible as a Telegram[14] bot: `https://t.me/ConvAI1337Bot`. Also we have public JSON API that documented at `https://github.com/sld/convai-bot-1337/wiki/Api-Documentation`. The source code is released under GNU GPLv3 license and available on GitHub: `https://github.com/sld/convai-bot-1337`.

Acknowledgements

We thank Mikhail Burtsev, Luiza Sayfullina and Mikhail Pavlov for comments that greatly improved the manuscript. We would also like to thank the Reason8.ai company for providing computational resources and grant for NIPS 2017 ticket. We thank Neural Systems and Deep Learning Lab of MIPT for ideas and support.

References

Jan Chorowski, Adrian Łańcucki, Szymon Malik, Maciej Pawlikowski, Paweł Rychlikowski, and Paweł Zykowski. 2018. A Talker Ensemble: the University of Wroclaw's Entry to the NIPS 2017 Conversational Intelligence Challenge. *arXiv preprint arXiv:1805.08032*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *CoRR*, abs/1603.08023.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. *CoRR*, abs/1605.05414.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126, Vancouver, Canada, July.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Michael L Mauldin. 1994. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *AAAI*, volume 94, pages 16–21.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA.

[14] `https://telegram.org/`

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational AI: The Science Behind the Alexa Prize. *arXiv preprint arXiv:1801.03604*.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical Neural Network Generative Models for Movie Dialogues. *CoRR*, abs/1507.04808.

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2009. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. 2015. Bigartm: Open source library for regularized multimodal topic modeling of large collections. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 370–381. Springer.

Richard S Wallace. 2009. The anatomy of ALICE. In *Parsing the Turing Test*, pages 181–210. Springer.

Joseph Weizenbaum. 1966. ELIZA - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural Question Generation from Text: A Preliminary Study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

Appendices

A Appendix: Dialog example

Text

In 2015-2016, Notre Dame ranked 18th overall among ”national universities” in the United States in U.S. News & World Report’s Best Colleges 2016. In 2014, USA Today ranked Notre Dame 10th overall for American universities based on data from College Factual. Forbes.com’s America’s Best Colleges ranks Notre Dame 13th among colleges in the United States in 2015 ... According to payscale.com, undergraduate alumni of University of Notre Dame have a mid-career median salary $110,000, making it the 24th highest among colleges and universities in the United States. The median starting salary of $55,300 ranked 58th in the same peer group.

Bot - Greeting skill: Hiya!
Bot - Greeting skill, Topic detection skill: Let's talk about culture, or I can ask you a question about this text.
User: Ask me then :)
Bot - Question asking skill: What is the median salary in notre dame?
User: 55,300
Bot - Answer checking skill: Show me your best! Hint: first 3 letters is $1. Try again.
User: What is the rank of Notre Dame?
Bot - Question answering skill: It seems that correct answer is: "18th overall among national universities in the united states in u.s. ..."
User: Nice!
Bot - Facebook news chit-chat skill: This is what happens when you don't want to be in the white house.
User: Very funny :) This is very long text to read, can you summarize it in couple sentences?
Bot - Text summarization skill: I suppose that this article main idea is: University of notre dame's salary.
User: Hope you are not lying to me
Bot - Facebook news chit-chat skill: This is what happens when you vote for a democrat.
User: You are so into politics
Bot - Alice chit-chat skill: I will tell my the kind people in the lab you said i am into politics.
User: I have to go! Bye!