

RE-WOCHAT: Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents - Development and Evaluation

Workshop Programme (May 28th, 2016)

13:50 – 14:00 – Welcome Message by the Organizing Team

14:00 – 15:00 – Short Paper Session I

Data Collection for Interactive Learning through the Dialog

Miroslav Vodolán, Filip Jurčíček

A Context-aware Natural Language Generation Dataset for Dialogue Systems

Ondřej Dušek, Filip Jurčíček

Dead Man Tweeting

David Nilsson, Magnus Sahlgren, Jussi Karlgren

15:00 – 16:00 – Short Paper Session II

Chatbot Evaluation and Database Expansion via Crowdsourcing

Zhou Yu, Ziyu Xu, Alan W. Black, Alexander I. Rudnicky

Framework for the Formulation of Metrics for Conversational Agent Evaluation

Mohammed Kaleem, Omar Alobadi, James O'Shea, Keeley Crockett

Tracking Conversational Gestures of Extraverts and Introverts in Multimodal Interaction

David Novick, Adriana Camacho, Ivan Gris, Laura M. Rodriguez

16:00 – 16:30 – Coffee break

16:30 – 17:00 – Long Paper Session

Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora

Bayan AbuShawar, Eric Atwell

17:00 – 17:50 – Shared Task Presentation and Poster Session

Shared Task on Data Collection and Annotation

Luis Fernando D'Haro, Bayan AbuShawar, Zhou Yu

Shared Task Chatbot Description Reports:

POLITICIAN, David Kuboň, Barbora Hladká

JOKER, Guillaume Dubuisson Duplessis, Vincent Letard, Anne-Laure Ligozat, Sophie Rosset

IRIS (Informal Response Interactive System), Rafael E. Banchs, Haizhou Li

PY-ELIZA (A Python-based implementation of Eliza), Luis Fernando D'Haro

TICK TOCK, Zhou Yu, Ziyu Xu, Alan W. Black, Alexander I. Rudnicky

SARAH, Bayan AbuShawar

17:50 – 18:00 – Concluding Remarks by the Organizing Team

Editors

Rafael E. Banchs	Institute for Infocomm Research, Singapore
Ryuichiro Higashinaka	Nippon Telegraph and Telephone Corporation, Japan
Wolfgang Minker	Ulm University, Germany
Joseph Mariani	IMMI & LIMSI-CNRS, France
David Traum	University of Southern California, USA

Workshop Organizers

Rafael E. Banchs	Institute for Infocomm Research, Singapore
Ryuichiro Higashinaka	Nippon Telegraph and Telephone Corporation, Japan
Wolfgang Minker	Ulm University, Germany
Joseph Mariani	IMMI & LIMSI-CNRS, France
David Traum	University of Southern California, USA

Shared Task Co-organizers

Bayan Abu Shawar	Arab Open University, Jordan
Luis Fernando D'Haro	Agency for Science, Technology and Research, Singapore
Zhou Yu	Carnegie Mellon University, USA

Workshop Programme Committee

Björn Schuller	Imperial College London, UK
David Suendermann	Educational Testing Service (ETS), USA
Diane Litman	University of Pittsburgh, USA
Dilek Hakkani-Tur	Microsoft Research, USA
Gabriel Skantze	KTH Royal Institute of Technology, Sweden
Haizhou Li	Institute for Infocomm Research, Singapore
Jason Williams	Microsoft Research, USA
Jiang Ridong	Agency for Science, Technology and Research, Singapore
Justine Cassell	Carnegie Mellon University, USA
Kristiina Jokinen	University of Helsinki, Finland
Kotaro Funakoshi	Honda Research Institute, Japan
Laurence Devillers	LIMSI-CNRS, France
Luisa Coheur	Lisbon University, Portugal
Matthew Henderson	Google Research, USA
Michael McTear	University of Ulster, UK
Mikio Nakano	Honda Research Institute, Japan
Nick Campbell	Trinity College Dublin, Ireland
Oliver Lemon	Heriot-Watt University, UK
Ramón López-Cózar	University of Granada, Spain
Sakriani Sakti	Nara Institute of Science and Technology, Japan
Satoshi Nakamura	Nara Institute of Science and Technology, Japan
Seokhwan Kim	Agency for Science, Technology and Research, Singapore
Sophie Rosset	LIMSI-CNRS, France
Stefan Ultes	Cambridge University, UK
Suraj Nair	Technische Universität München, Germany
Teruhisa Misu	Honda Research Institute, USA
Tomoki Toda	Nara Institute of Science and Technology, Japan
Yasuharo Den	Chiba University, Japan
Shiv Vitaladevuni	Amazon, USA

Table of contents

<i>Introduction to RE-WOCHAT</i>	v
Rafael E. Banchs, Ryuichiro Higashinaka, Wolfgang Minker, Joseph Mariani, David Traum	
Short Papers	
<i>Data Collection for Interactive Learning through the Dialog</i>	1
Miroslav Vodolán, Filip Jurčiček	
<i>A Context-aware Natural Language Generation Dataset for Dialogue Systems</i>	6
Ondřej Dušek, Filip Jurčiček	
<i>Dead Man Tweeting</i>	10
David Nilsson, Magnus Sahlgren, Jussi Karlgren	
<i>Chatbot Evaluation and Database Expansion via Crowdsourcing</i>	15
Zhou Yu, Ziyu Xu, Alan W. Black, Alexander I. Rudnicky	
<i>Framework for the Formulation of Metrics for Conversational Agent Evaluation</i>	20
Mohammed Kaleem, Omar Alobadi, James O’Shea, Keeley Crockett	
<i>Tracking Conversational Gestures of Extraverts and Introverts in Multimodal Interaction</i> . . .	24
David Novick, Adriana Camacho, Ivan Gris, Laura M. Rodriguez	
Long Papers	
<i>Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora</i>	29
Bayan AbuShawar, Eric Atwell	
Shared Task Reports	
<i>Shared Task on Data Collection and Annotation</i>	39
Luis Fernando D’Haro, Bayan AbuShawar, Zhou Yu	
<i>Politician</i>	43
David Kuboň, Barbora Hladká	
<i>Joker Chatterbot</i>	45
Guillaume Dubuisson Duplessis, Vincent Letard, Anne-Laure Ligozat, Sophie Rosset	
<i>IRIS (Informal Response Interactive System)</i>	47
Rafael E. Banchs, Haizhou Li	
<i>Py-Eliza: A Python-based Implementation of the Famous Computer Therapist</i>	49
Luis Fernando D’Haro	
<i>TickTock</i>	51
Zhou Yu, Ziyu Xu, Alan W. Black, Alexander I. Rudnicky	
<i>Sarah Chatbot</i>	53
Bayan AbuShawar	

Author Index

A

- AbuShawar, Bayan. 29, 39, 53
Alobadi, Omar. 20
Atwell, Eric. 29

B

- Banchs, Rafael E. *v*, 47
Black, Alan W. 15, 51

C

- Camacho, Adriana. 24
Crockett, Keeley 20

D

- D'Haro, Luis F. 39, 49
Dubuisson, Guillaume. 45
Dušek, Ondřej. 6

G

- Gris, Ivan. 24

H

- Higashinaka, Ryuichiro. *v*
Hladká, Barbora 43

J

- Jurčiček, Filip 1, 6

K

- Kaleem, Mohammed. 20
Karlgrén, Jussi. 10
Kuboň, David 43

L

- Li, Haizhou 47
Ligozat, Anne-Laure 45
Letard, Vincent 45

M

- Mariani, Joseph *v*
Minker, Wolfgang. *v*

N

- Nilsson, David. 10
Novick, David 24

O

- O'Shea, James. 20

R

- Rodriguez, Laura M. 24
Rosse, Sophie 45
Rudnicky, Alexander I. 15, 51

S

- Sahlgren, Magnus 10

T

- Traum, David. *v*

V

- Vodolán, Miroslav. 1

X

- Xu, Ziyu. 15, 51

Y

- Yu, Zhou 15, 39, 51

Introduction to RE-WOCHAT

Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents – Development and Evaluation

Although non-goal-oriented dialogue systems have been around for many years (more than forty years indeed, if we consider Weizenbaum’s Eliza system as the starting milestone), they have been recently gaining a lot of popularity in both research and commercial arenas. From the commercial stand point, non-goal-oriented dialogue seems to be providing an excellent means to engage users for entertainment purposes, as well as to give a more human-like appearance to established vertical goal-driven dialogue systems.

From the research perspective, on the other hand, this kind of systems poses interesting challenges and problems to the research community. Different from goal-oriented dialogue systems, which are domain focused, non-goal-oriented or chat-oriented dialogue requires dealing with knowledge on a wide diversity of domains as well as common sense knowledge related to daily life experiences. Additionally, due to the lack of specific goals in chat-oriented dialogue, this kind of systems cannot be objectively evaluated by using goal completion rates, as in the case of goal-oriented dialogue engines. Moreover, many task-oriented dialogue systems use length of the dialogue as a metric, with a penalty in reward function or assumed user-satisfaction for longer dialogues. For chat dialogue, however, this metric is often reversed: the more interested and enjoyable the chat dialogue is, the longer users will talk with a system.

In this regards, the RE-WOCHAT initiative aims at providing and consolidating a venue for the research community to explore and discuss the state-of-the-art in non-goal-oriented dialogue and its related problems, including resource generation and evaluation. The workshop also accommodates a shared task on “Data Collection and Annotation” aiming at developing and testing a new evaluation framework for non-goal-oriented dialogue engines.

RE-WOCHAT is the result of a working committee initiative on “Automatic Evaluation and Resources” generated during the Shonan Meeting “The Future of Human-Robot Spoken Dialogue: from Information Services to Virtual Assistants” held in Shonan, Japan, at the end of March 2015. The main objective of this meeting was to discuss about the most relevant and promising future directions of research in dialogue systems. The discussion was centred on how these directions should address the different problems and limitations of current dialogue systems, as well as how they can provide the basis for the next generation of intelligent artificial agents. (More information is available at <http://shonan.nii.ac.jp/shonan/wp-content/uploads/2011/09/No.2015-7.pdf>).

The workshop also constitutes the natural extension of two successful Special Sessions on “Chatbots and Dialogue Agents” collocated with APSIPA conferences in 2014 and 2015 (<http://www.apsipa2014.org/home/program/special-sessions> and <http://www.apsipa2015.org/>), and two evaluation workshops related to chat-oriented dialogue systems: dialogue breakdown detection challenge (<https://sites.google.com/site/dialoguebreakdowndetection/>) and NTCIR short text conversation (<http://ntcir12.noahlab.com.hk/stc.htm>).

Rafael E. Banchs, Ryuichiro Higashinaka, Wolfgang Minker, Joseph Mariani, David Traum
Portorož, Slovenia, May 28th, 2016

Data Collection for Interactive Learning through the Dialog

Miroslav Vodolán, Filip Jurčiček

Charles University in Prague, Faculty of Mathematics and Physics

Institute of Formal and Applied Linguistics

Malostranské náměstí 25, 11800 Praha 1, Czech Republic

{vodolan, jurcicek}@ufal.mff.cuni.cz

Abstract

This paper presents a dataset collected from natural dialogs which enables to test the ability of dialog systems to learn new facts from user utterances throughout the dialog. This *interactive learning* will help with one of the most prevailing problems of open domain dialog system, which is the sparsity of facts a dialog system can reason about. The proposed dataset, consisting of 1900 collected dialogs, allows simulation of an interactive gaining of denotations and questions explanations from users which can be used for the *interactive learning*.

Keywords: dataset, data collection, dialog, knowledge graph, interactive learning

1. Introduction

Nowadays, dialog systems are usually designed for a single domain (Mrksic et al., 2015). They store data in a well-defined format with a fixed number of attributes for entities that the system can provide. Because data in this format can be stored as a two-dimensional table within a relational database, we call the data flat. This data representation allows the system to query the database in a simple and efficient way. It also allows to keep the dialog state in the form of slots (which usually correspond to columns in the table) and track it through the dialog using probabilistic belief tracking (Williams et al., 2013; Henderson et al., 2014).

However, the well-defined structure of the database of a typical dialog system comes with a high cost of extending it as every piece of new information has to fit the format. This is especially a problem when we one is adapting the system for a new domain because its entities could have different attributes.

A dialog system based on knowledge bases offers many advantages. First, the knowledge base, which can be represented as knowledge graph containing entities connected by relations, is much more flexible than the relational database. Second, freely available knowledge bases, such as Freebase, Wikidata, etc. contain an enormous amount of structured information, and are still growing. A dialog system which is capable of working with this type of information would be therefore very useful.

In this paper we propose a dataset aiming to help develop and evaluate dialog systems based on knowledge bases by *interactive learning* motivated in Section 2. Section 3. describes policies that can be used for retrieving information from knowledge bases. In Section 4. is introduced a dialog simulation from natural conversations which we use for evaluation of *interactive learning*. The dataset collection process allowing the dialog simulation is described in Section 5. and is followed by properties of the resulting dataset in Section 6. Evaluation guidelines with proposed metrics can be found in Section 7. The planned future work is summarized in Section 8. We conclude the paper with Section 9.

2. Motivation

From the point of view of dialog systems providing general information from a knowledge base, the most limiting factor is that a large portion of the questions is understood poorly.

Current approaches (Berant and Liang, 2015; Bordes et al., 2014) can only achieve around 50% accuracy on some question answering datasets. Therefore, we think that there is a room for improvements which can be achieved by interactively asking for additional information in conversational dialogs with users. This extra information can be used for improving policies of dialog systems. We call this approach the *interactive learning* from dialogs.

We can improve dialog systems in several aspects through *interactive learning* in a direct interaction with users. First, the most straightforward way obviously is getting the correct answer for questions that the system does not know. We can try to ask users for answers on questions that the system encountered in a conversation with a different user and did not understand it. Second, the system can ask the user for a broader explanation of a question. This explanation could help the system to understand the question and provide the correct answer. In addition, the system can learn correct policy for the question which allows providing answers without asking any extra information for similar questions next time. We hypothesize that users are willing to give such explanations because it could help them to find answers for their own questions. The last source of information that we consider for *interactive learning* is rephrasing, which could help when the system does know the concept but does not know the correct wording. This area is extensively studied for the purposes of information retrieval (Imielinski, 2009; France et al., 2003).

The main purpose of the collected dataset is to enable *interactive learning* using the steps proposed above and potentially to evaluate how different systems perform on this task.

3. Dialog policies

The obvious difficulty when developing a dialog system is finding a way how to identify the piece of information that

the user is interested in. This is especially a problem for dialog systems based on knowledge graphs containing a large amount of complex structured information. While a similar problem is being solved in a task of question answering, dialog systems have more possibilities of identifying the real intention of the user. For example, a dialog system can ask for additional information during the dialog.

We distinguish three different basic approaches to requesting knowledge bases:

handcrafted policy – the policy consists of fixed set of rules implemented by system developers,

offline policy – the policy is learned from some kind of offline training data (usually annotated) without interaction with system users (Bordes et al., 2015),

interactively learned policy – the system learns the policy through the dialog from its users by interactively asking them for additional information.

A combination of the above approaches is also possible. For example, we can imagine scenarios where the dialog system starts with hand-crafted rules, which are subsequently interactively improved through dialogs with its users. With a growing demand for open domain dialog systems, it shows that creating hand-crafted policies does not scale well - therefore, machine learning approaches are gaining on popularity. Many public datasets for offline learning have been published (Berant et al., 2013; Bordes et al., 2015). However, to our knowledge, no public datasets for interactive learning are available. To fill this gap, we collected a dataset which enables to train interactively learned policies through a simulated interaction with users.

4. Dialog Simulation

Offline evaluation of interactive dialogs on real data is difficult because different policies can lead to different variants of the dialog. Our solution to this issue is to collect data in a way that allows us to simulate all dialog variants possible according to any policy.

The dialog variants we are considering for *interactive learning* differ only in presence of several parts of the dialog. Therefore, we can collect dialogs containing all information used for interactive learning and omit those parts that were not requested by the policy.

We collected the dataset (see Section 5.) that enables simulation where the policy can decide how much extra information to the question it requests. If the question is clear to the system it can attempt to answer the question without any other information. It can also ask for a broader explanation with a possibility to answer the question afterwards. If the system decides not to answer the question, we can simulate rerouting the question to another user, to try to obtain the answer from them. The principle of simulated user’s answer is shown in the Figure 1.

Note that the simulated user’s answer can be incorrect because human users naturally made mistakes. We intentionally keep these mistakes in the dataset because real systems must address them as well.

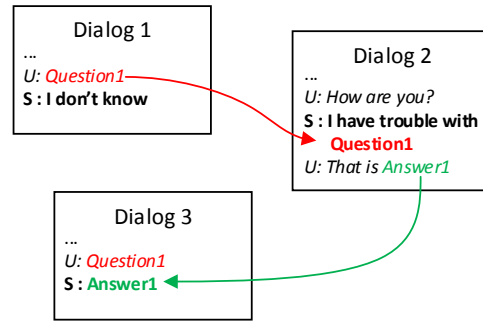


Figure 1: Unknown questions can be rerouted between users. We can, for example, use chitchat to get correct answers. The challenge is in generalizing the collected question-answer pairs using the knowledge base in order to apply them to previously unseen questions.

5. Dataset Collection Process

A perfect data collection scenario for our dataset would use real running dialog system providing general information from the knowledge base to real users. This system could then ask for explanations and answers for questions which it is not able to answer.

However, getting access to systems with real users is usually hard. Therefore, we used the crowdsourcing platform CrowdFlower¹ (CF) for our data collection.

A CF worker gets a task instructing them to use our chat-like interface to help the system with a question which is randomly selected from training examples of Simple questions (Bordes et al., 2015) dataset. To complete the task user has to communicate with the system through the three phase dialog discussing question paraphrase (see Section 5.1.), explanation (see Section 5.2.) and answer of the question (see Section 5.3.). To avoid poor English level of dialogs we involved CF workers from English speaking countries only. The collected dialogs has been annotated (see Section 5.4.) by expert annotators afterwards.

The described procedure leads to dialogs like the one shown in the Figure 2.

5.1. Question Paraphrasing

At beginning of the dialog, the system is requesting the user to paraphrase question that the system does not understand. The main goal of this first phase is to let the user get familiar with the presented question and to get alternative wordings of the posed question.

5.2. Question Explanation

In the second phase, the user is asked for an explanation of the question. We expect the explanation to be different enough from the original question (in terms of the number of common words between the question and the explanation). If the explanation is too similar to the question, the user is notified that their explanation is not broad enough and they must provide a better one.

¹<http://crowdfLOWER.com>

S1: Hello, I need help with this question: 'what country is pilar mateos from?'

Can you put this question in a different way?

U1: We are wondering what is the nationality of a specific person.

S2: This seems like an interesting question.

Could you explain in detail what it is really asking for?

U2: The nationality is the origine, the country where someone is born.

S3: It sounds reasonable, however, I still cannot find the answer. Could you give me the correct answer for the question please?

U3: The answer is Spain, Pilar Mateos is Spanish.

Figure 2: An Example of a short dialog collected on the crowdsourcing platform. We can see that the user provides the question paraphrase (S1), the explanation (S2) and the correct answer for the question (S3).

5.3. Question Answer

With the valid explanation the dialog turns into the last phase where the user is asked for a correct answer to the original question. The system requires the user to answer with a full sentence. In practical experiments this has shown as a useful decision because it improves system’s ability to reveal cheaters. We can simply measure the connection (in terms of common words) between question and the answer sentence. This allows to reject completely irrelevant answers.

5.4. Annotation

The correct answer for question in each dialog is available from Simple questions dataset. Answers are in form of Freebase² entities identified by unique id. For evaluation purposes we need information whether dialog contains the answer which is consistent with the entity from Simple questions, the answer with another entity or whether the dialog does not contain any answer. While the annotation process is quite simple, we did not need crowdsourcing for the process.

5.5. Natural Language Understanding (NLU)

The collection system needs to recognize following dialog acts from user utterances during all phases of the dialog:

Negate – user does not want to provide requested information,

Affirm – user agrees to provide requested information,

DontKnow – user does not know the requested information,

ChitChat – user tries chit chat with the system (hello, bye, who are you...),

Inform – none of the above, interpreted as user is giving information requested by the system.

Parsing of the dialog acts is made by hand written rules using templates and keyword spotting. The templates and keywords were manually collected from frequent expressions used by CF workers during preparation runs of the dataset collection process (google it, check wikipedia, I would need... → Negate).

6. Dataset Properties

We collected the dataset with 1900 dialogs and 8533 turns. Topics discussed in dialogs are questions randomly chosen from training examples of Simple questions (Bordes et al., 2015) dataset. From this dataset we also took the correct answers in form of Freebase entities.

Our dataset consists of standard data split into training, development and test files. The basic properties of those files are as follows:

	dialog count	dialog turns
Training dialogs	950	4249
Development dialogs	285	1258
Testing dialogs	665	3026

Table 1: Table of turn and dialog counts for dataset splits.

Each file contains complete dialogs enriched by outputs of NLU (see Section 5.5.) that were used during the data collection. On top of that, each dialog is labeled by the correct answer for the question and expert annotation of the user answer hint which tells whether the hint points to the correct answer, incorrect answer, or no answer at all.

351 of all collected dialogs contain correct answer provided by users and 702 dialogs have incorrect answer. In the remaining 847 dialogs users did not want to answer the question. The collected dialogs also contain 1828 paraphrases and 1539 explanations for 1870 questions.

An answer for a question was labeled as correct by annotators only when it was evident to them that the answer points to the same Freebase entity that was present in Simple questions dataset for that particular question. However, a large amount of questions from that dataset is quite general - with many possible answers. Therefore lot of answers from users were labeled as incorrect even though those answers perfectly fit the question. Our annotators identified that 285 of the incorrect answers were answers for such general questions. Example of this situation can be demonstrated by question '*Name an actor*' which was correctly answered by '*Brad Pitt is an actor*', however, to be consistent with Simple questions annotation, which is '*Kelly Atwood*', annotators were forced to mark it as an incorrect answer.

7. Interactive Learning Evaluation

A perfect *interactive learning* model would be able to learn anything interactively from test dialogs during testing, which would allow us to measure progress of the model

²<https://www.freebase.com/>

from scratch over the course of time. However, a development of such model would be unnecessarily hard, therefore we provide training dialogs which can be used for feature extraction and other engineering related to *interactive learning* from dialogs in natural language. Model development is further supported with labeled validation data for parameter tuning.

We propose two evaluation metrics for comparing *interactive learning* models. First metric (see Section 7.1.) scores amount of information required by the model, second metric (see Section 7.2.) is accuracy of answer extraction from user utterances. All models must base their answers only on information gained from training dialogs and testing dialogs seen during the simulation so far, to ensure that the score will reflect the *interactive learning* of the model instead of general question answering.

7.1. Efficiency Score

The simulation of dialogs from our dataset allows to evaluate how efficient a dialog system is in using information gained from users. The dialog system should maximize the number of correctly answered questions without requesting too many explanations and answers from users. To evaluate different systems using the collected data, we propose the following evaluation measure:

$$S_D = \frac{n_c - w_i n_i - w_e n_e - w_a n_a}{|D|} \quad (1)$$

Here, n_c denotes the number of correctly answered questions, n_i denotes the number of incorrectly answered questions, n_e denotes the number of requested explanations, n_a denotes the number of requested answers and $|D|$ denotes the number of simulated dialogs in the dataset. w_i, w_e, w_a are penalization weights.

The penalization weights are used to compensate for different costs of obtaining different types of information from the user. For example, gaining broader explanation from the user is relatively simple because it is in their favor to cooperate with the system on a question they are interested in. However, obtaining correct answers from users is significantly more difficult because the system does not always have the chance to ask the question and the user does not have to know the correct answer for it.

To make the evaluations comparable between different systems we recommend using our evaluation scripts included with the dataset with following penalization weights that reflect our intuition for gaining information from users:

- $w_i = 5$ – incorrect answers are penalized significantly,
- $w_e = 0.2$ – explanations are quite cheap; therefore, we will penalize them just slightly,
- $w_a = 1$ – gaining question’s answer from users is harder than gaining explanations.

7.2. Answer Extraction Accuracy

It is quite challenging to find appropriate entity in the knowledge base even though the user provided the correct answer. Therefore, we propose another metric relevant to our dataset. This metric is the accuracy of entity extraction

which measures how many times was extracted a correct answer from answer hints provided by the user in dialogs annotated as correctly answered.

8. Future Work

Our future work will be mainly focused on providing a baseline system for interactive learning which will be evaluated on the dataset. We are also planning improvements for dialog management that is used to gain explanations during the data collection. We believe that with conversation about specific aspects of the discussed question it will be possible to gain even more interesting information from users. The other area of our interest is in possibilities to improve question answering accuracy on test questions of Simple question dataset with the extra information contained in the collected dialogs.

9. Conclusion

In this paper, we presented a novel way how to evaluate different interactive learning approaches for dialog models. The evaluation covers two challenging aspects of interactive learning. First, it scores efficiency of using information gained from users in simulated question answering dialogs. Second, it measures accuracy on answer hints understanding.

For purposes of evaluation we collected a dataset from conversational dialogs with workers on crowdsourcing platform CrowdFlower. Those dialogs were annotated with expert annotators and published under Creative Commons 4.0 BY-SA license on lindat³. We also provide evaluation scripts with the dataset that should ensure comparable evaluation of different interactive learning approaches.

10. Acknowledgments

This work was funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221 and core research funding, SVV project 260 224, and GAUK grant 1170516 of Charles University in Prague. It used language resources stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

11. Bibliographical References

- Berant, J. and Liang, P. (2015). Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics (ACL)*, 3:545–558.
- Berant, J., Chou, A., Frostig, R., and Liang, P. (2013). Semantic Parsing on Freebase from Question-Answer Pairs. *Proceedings of EMNLP*, (October):1533–1544.
- Bordes, A., Chopra, S., and Weston, J. (2014). Question Answering with Subgraph Embeddings.
- Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale Simple Question Answering with Memory Networks.

³hdl.handle.net/11234/1-1670

- France, F. D., Yvon, F., and Collin, O. (2003). Learning paraphrases to improve a question-answering system. In *In Proceedings of the 10th Conference of EACL Workshop Natural Language Processing for Question-Answering*.
- Henderson, M., Thomson, B., and Williams, J. (2014). The second dialog state tracking challenge. In *15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, volume 263.
- Imielinski, T. (2009). If you ask nicely , I will answer : Semantic Search and Today's Search Engines. *Search*.
- Mrksic, N., Séaghdha, D. Ó., Thomson, B., Gasic, M., Su, P., Vandyke, D., Wen, T., and Young, S. J. (2015). Multi-domain dialog state tracking using recurrent neural networks. *CoRR*, abs/1506.07190.
- Williams, J., Raux, A., Ramachandran, D., and Black, A. (2013). The Dialog State Tracking Challenge. *Sigdialog*, (August):404–413.

A Context-aware Natural Language Generation Dataset for Dialogue Systems

Ondřej Dušek, Filip Jurčiček

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 11800 Praha 1, Czech Republic
{odusek,jurcicek}@ufal.mff.cuni.cz

Abstract

We present a novel dataset for natural language generation (NLG) in spoken dialogue systems which includes preceding context (user utterance) along with each system response to be generated, i.e., each pair of source meaning representation and target natural language paraphrase. We expect this to allow an NLG system to adapt (entrain) to the user’s way of speaking, thus creating more natural and potentially more successful responses. The dataset has been collected using crowdsourcing, with several stages to obtain natural user utterances and corresponding relevant, natural, and contextually bound system responses. The dataset is available for download under the Creative Commons 4.0 BY-SA license.

Keywords: natural language generation, entrainment, task-oriented dialogue

1. Introduction

We present a new dataset intended for fully trainable natural language generation (NLG) systems in task-oriented spoken dialogue systems (SDS). It is, to our knowledge, the first dataset of its kind to include preceding context (user utterance) with each data instance (source meaning representation and target natural language paraphrase to be generated, see Figure 1). Taking the form of the previous user utterance into account for generating the system response should presumably improve the perceived naturalness of the output, and may even lead to a higher task success rate (see Section 3.). Crowdsourcing has been used to obtain natural context user utterances as well as natural system responses to be generated. The dataset covers the domain of public transport information and is released under a permissive Creative Commons 4.0 BY-SA license.

NLG systems in current SDS are in most cases handcrafted, e.g., (Rudnicky et al., 1999; Raux et al., 2005). Such systems are efficient and maintainable for limited domains, but provide little to no variance in their outputs, which makes them repetitive. Their scalability is also limited (Mairesse and Walker, 2011). Recent fully trainable NLG systems for SDS typically use random sampling to provide variance in outputs (Mairesse et al., 2010; Wen et al., 2015a; Wen et al., 2015b). This is perceived as more natural by the users, but still lacks adaptation to previous context, which is the norm in human-human dialogues.

We believe that the present dataset can be used for proof-of-concept experiments studying context adaptation in human-computer dialogues and that the results will be applicable to other domains as well as open-domain and chat-oriented systems. The method used to collect the data is completely domain-independent.

This paper is structured as follows: Section 2. introduces the task of NLG in SDS and describes the dialogue system and domain used in data collection. We give a brief explanation of the phenomenon of dialogue alignment, or entrainment, in Section 3. Section 4. then contains a description of the data collection process. We outline the main properties of the dataset in Section 5., and we list related works in Section 6. Section 7. then concludes the paper.

```
inform(vehicle=subway,line=C,  
      from_stop=Bowery,to_stop=Central Park,  
      departure_time=10:04am)
```

OK, take the C subway from Bowery heading for Central Park at 10:04am.

Figure 1: An example of NLG input (top) and output (bottom) in a task-oriented SDS

2. Natural Language Generation in Task-oriented Spoken Dialogue Systems

We understand the task of NLG in the context of task-oriented SDS which use *dialogue acts* (DA) to represent meaning (Young et al., 2010; Jurčiček et al., 2014). A DA represents a specific system or user action, such as *hello*, *inform*, *confirm*, or *request*. It is typically accompanied by one or more *slots* (variables) which may take specific values. The job of NLG in this context is to translate an input DA into one or more sentences in a natural language. An example input-output pair is shown in Figure 1.

We use the domain of English public transport information as implemented in the Alex SDS framework (Jurčiček et al., 2014; Vejman, 2015). It is a mixed-initiative dialogue system using Google Maps API to find public transit directions among bus and subway stops on Manhattan.¹ The user is able to specify a time preference or select a means of transport; they may ask for duration or trip distance.

3. Entrainment in Dialogue

Entrainment in dialogue, also referred to as alignment or adaption, is the mutual linguistic convergence of speakers during the course of a conversation. Speakers are primed (influenced) by previous utterances (Reitter et al., 2006) and tend to reuse vocabulary, syntactic structure, and prosody (Levitan, 2014) (see Figure 3). Entrainment occurs naturally and subconsciously and facilitates successful conversations (Friedberg et al., 2012).

¹The Alex system handles a larger domain, but we limited it to prevent data sparsity when collecting our dataset.

Nenkova et al. (2008) have shown that higher entrainment in frequent words correlates with a higher success rate in task-oriented human-human dialogues. Users have been reported to entrain naturally to prompts of a SDS (Stoyanchev and Stent, 2009; Parent and Eskenazi, 2010).

There have been several attempts to introduce a two-way entrainment into SDS, i.e., let the system entrain to user utterances. Hu et al. (2014) report an increased naturalness of the system responses, while Lopes et al. (2013) and Lopes et al. (2015) also mention increased task success. All of these approaches focus on lexical entrainment and are completely or partially rule-based.

Using the present dataset, we are planning to take entrainment even further in the context of a fully trainable NLG and train a system that adapts to users' lexical as well as syntactic choices. We hope that this will further increase both perceived naturalness of the system responses and overall task success rate.

4. Dataset Collection Process

When collecting the dataset, we aimed at capturing naturally occurring entrainment between pairs of user utterances and system responses. Collecting complete natural human-human task-oriented dialogues would probably yield better conditions for entrainment and make much wider contexts available in our dataset. However, in order to avoid data sparsity, we limited the context to a single preceding user utterance, which is likely to have the largest entrainment influence.

To obtain both natural user utterances and natural system responses, we took the following approach: First, user utterances were recorded in calls to a live SDS (see Section 4.1.). The recorded utterances were then transcribed (see Section 4.2.), and the transcriptions were parsed and delexicalized (see Section 4.3.). Finally, based on the meaning of the user utterances, we generated possible response DA (see Section 4.4.) and obtained their natural language paraphrases (see Section 4.5.).

We used the CrowdFlower (CF) platform² to crowdsource call recording, transcription, and response paraphrase creation. To attract native speakers only, the tasks were only made available to CF users in English-speaking countries.

4.1. Recording Calls

Using the Alex English public transport information SDS (Vejman, 2015), we recorded calls in a setting similar to SDS user evaluation (Jurčiček et al., 2011).³ CF users were given tasks that they should attempt to achieve with the system running on a toll-free phone number. The SDS would give them a code that allows them to collect CF reward.

The task descriptions presented to the users were designed so that variable and natural utterances are obtained. Even though the task itself stayed relatively similar,⁴ we varied

the description and used different synonyms (e.g., *schedule/ride/connection*) so that the users are primed with varying expressions. To generate the task descriptions, we used the Alex template NLG system with a specially-designed set of templates where many combinations can be created at random. Furthermore, the users were not aware that the exact wording of their requests is important. According to manual cursory checks of the recordings, they mostly tried to complete the task assigned to them and often kept to wording given to them in the description.

We collected 177 calls comprising 1,636 user utterances. We decided to also include recordings collected previously by Vejman (2015) (347 calls and 2,530 utterances). The response generation step (see Section 4.4.) selected 630 relevant utterances from our calls and 384 utterances from the calls of Vejman (2015).

4.2. Transcription

To ensure that the context user utterances in our dataset are accurate, we had our recorded calls manually transcribed using the standard CF transcription task. A brief description of the domain and lists of frequent words/expressions and subway stations were provided to transcribers to minimize the number of errors.

We collected three transcriptions per utterance and used the transcription variant provided by at least two users, resolving a small number of problematic cases manually.

4.3. Re-parsing

We needed to identify the meaning of the transcribed user utterances in order to generate relevant system response DA (see Section 4.4.). While the recorded calls contain Spoken Language Understanding (SLU) parses of all user utterances, those are based on speech recognition transcriptions. We applied the rule-based Alex SLU system again to manual transcriptions in order to obtain more reliable parses.

To reduce data sparsity, we delexicalized the utterances based on their SLU parses – all stop names as well as time expressions and names of transport vehicles were replaced with placeholders. Identical delexicalized utterances are treated as a single utterance (one context instance) in the dataset, but the frequency information is retained.

4.4. Generating response DA

We have created a simple rule-based bigram policy to generate all possible system response DA.⁵ Based on the given user utterance, it can generate several types of responses:

- a confirmation that the system understands the utterance (DA type `confirm`),
- an answer, providing a transport connection or specific details (DA type `inform`),
- an apology stating that the specified connection cannot be found (DA type `inform_no_match`),
- a request for additional information to complete search (DA type `request`).

The `confirm` response may further be combined with `inform` or `request` in a single utterance. As our policy is

⁵In a real dialogue, the correct response would depend on the whole dialogue history.

²<http://crowdflower.com>

³The task design was adapted from Vejman (2015).

⁴The users were supposed to ask for directions between two stops and request several additional details, such as duration of the ride, or ask for a schedule at a different time.

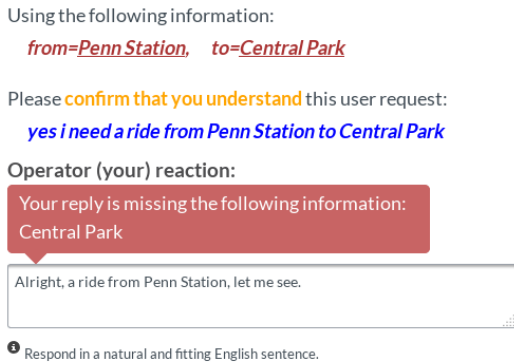


Figure 2: A response task in the CrowdFlower interface

only able to react to our limited domain (see Section 2.), it implicitly filters out all irrelevant user utterances.

4.5. Obtaining response paraphrases for NLG

The generated response DA were then used as the input to a CF task (see Figure 1) where users were asked to create appropriate natural language paraphrases. We designed the CF task interface iteratively based on several trial runs.

The CF user is asked to write a response of a certain kind (corresponding to DA types listed in Section 4.4.) and given information (slots and values) to back it up. The context user utterance is displayed directly above the text entry area to maximize entrainment influence. This simulates a natural situation where a hotline operator hears a request and responds to it immediately. To avoid priming CF users with slot names (e.g., `from_stop`, `departure_time`), we left out slot names where the meaning is unambiguous from the value (e.g., in time expressions) and used very short descriptions (e.g., `from`, `to`) elsewhere.⁶ The task instructions are relatively short and do not include any response examples so that CF users are not influenced by them.⁷

We use a JavaScript checker directly within the CF task to ensure that the paraphrase contains all required information (the exact value for stop names or time, or one of several synonyms in other slots). We also check for presence of irrelevant information, such as stop names, time expressions, or transport vehicles not included in the assignment.⁸ To check the created responses for fluency, we use AJAX calls to our spell-checking server based on Hunspell.⁹

Since about 20% of the responses collected in the testing runs contained errors (irrelevant information or non-fluent responses not discovered by our checks), we performed a manual quality control of all collected responses and requested additional paraphrases on CF where needed. This is quite straightforward and manageable given the size of our dataset; for larger datasets, crowdsourcing could also be used in quality control (Mitchell et al., 2014).

⁶We experimented with using pictographs instead of textual descriptions, but they proved to be rather confusing to CF users.

⁷A testing run with response examples did not bring a better quality of the responses.

⁸In our testing runs, CF users would often fabulate irrelevant information and include it in their responses.

⁹<http://hunspell.github.io>

total response paraphrases	5,577
unique (delex.) context + response DA	1,859
unique (delex.) context	552
unique (delex.) context with min. 2 occurrences	119
unique response DA	83
unique response DA types	6
unique slots	13

Table 1: Dataset size statistics

DA	count
<code>inform_no_match</code>	380
<code>iconfirm</code>	403
<code>iconfirm&inform</code>	23
<code>iconfirm&request</code>	252
<code>inform</code>	549
<code>request</code>	252

Table 2: System response DA counts in the dataset

5. Dataset Properties

The dataset was created over the course of three months, with an estimated net data collection time of one month. The final size statistics are shown in Table 1. There are 1,859 pairs of (delexicalized) context user utterances and system response DA in total, with three natural language paraphrases per pair. The set contains 83 different system response DA, which is lower than similar NLG datasets (see Section 6.), but sufficient to cover our domain. The 552 distinct context utterances provide ample space for entrainment experiments. Based on an estimate measured on a portion of the collected data, around 59% response paraphrases are syntactically aligned to context utterances, around 31% reuse their lexical items, and around 19% show both behaviors (see Figure 3). Statistics of the different DA types used in the dataset are given in Table 2.

The dataset is released in CSV and JSON formats and includes the following for each of the 1,859 items:

- context user utterance
- occurrence count of the user utterance in recorded calls
- SLU parse of the user utterance
- generated system response DA
- 3 natural language paraphrases of the system response

6. Related Work

Other publicly available datasets known to us which are specifically designed for NLG in SDS are those by Mairesse et al. (2010) and Wen et al. (2015b). Both works involve a restaurant information domain, the latter provides an additional set covering hotels. All sets have been obtained using crowdsourcing and contain around 200 distinct system response DA, with ca. 400 paraphrases in the former and around 5,000 in the latter case, which is comparable to our set. None of the sets include context user utterances.

Also related to our work are large-scale datasets of unstructured dialogues (cf. the survey of Serban et al. (2015, p. 21)). They are an order of magnitude larger than our dataset and include up to a full dialogue history, but they contain no semantic annotation, provide no explicit way of controlling the dialogue flow, and are not directly applicable to task-oriented SDS.

context utterance	response DA	response paraphrase
<i>how bout the next ride</i>	<code>inform_no_match(alternative=next)</code>	Sorry, I did not find a later option. I'm sorry, <u>the next ride</u> was not found.
<i>what is the distance of this trip</i>	<code>inform(distance=10.4 miles)</code>	<u>The distance is</u> 10.4 miles. <u>It is</u> around 10.4 miles. The <u>trip</u> covers a <u>distance</u> of 10.4 miles.

Figure 3: Entrainment examples from our dataset (entraining elements marked in color: lexical, syntactic, both).

7. Conclusion

We have presented a novel NLG dataset for the dialogue covering the domain of English public transport information, along with the method to obtain the data using crowd-sourcing. It is, to our knowledge, the first publicly available dataset applicable to experiments with entrainment, or dialogue alignment, in a SDS. The dataset is released under the Creative Commons 4.0 BY-SA license at the following URL:¹⁰

<http://hdl.handle.net/11234/1-1675>

We intend to use the dataset with a fully trainable NLG system in the Alex SDS (Jurčiček et al., 2014) and evaluate perceived naturalness of system responses.

8. Acknowledgments

This work was funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221 and core research funding, SVV project 260 224, and GAUK grant 2058214 of Charles University in Prague. It used language resources stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

9. Bibliographical References

- Friedberg, H., Litman, D., and Paletz, S. B. (2012). Lexical entrainment and success in student engineering groups. In *Proc. of SLT*, pages 404–409.
- Hu, Z., Halberg, G., Jimenez, C., and Walker, M. (2014). Entrainment in pedestrian direction giving: How many kinds of entrainment. In *Proc. IWSDS*, pages 90–101.
- Jurčiček, F., Keizer, S., Gašić, M., Mairesse, F., Thomson, B., Yu, K., and Young, S. (2011). Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk. In *Proc. of Interspeech*, pages 3068–3071.
- Jurčiček, F., Dušek, O., Plátek, O., and Žilka, L. (2014). Alex: A statistical dialogue systems framework. In *Proc. of Text, Speech and Dialogue*, pages 587–594.
- Levitan, R. (2014). *Acoustic-Prosodic Entrainment in Human-Human and Human-Computer Dialogue*. Ph.D. thesis, Columbia University.
- Lopes, J., Eskenazi, M., and Trancoso, I. (2013). Automated two-way entrainment to improve spoken dialog system performance. In *ICASSP*, pages 8372–8376.
- Lopes, J., Eskenazi, M., and Trancoso, I. (2015). From rule-based to data-driven lexical entrainment models in spoken dialog systems. *Computer Speech & Language*, 31(1):87–112.
- Mairesse, F. and Walker, M. A. (2011). Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488.
- Mairesse, F., Gašić, M., Jurčiček, F., Keizer, S., Thomson, B., Yu, K., and Young, S. (2010). Phrase-based statistical language generation using graphical models and active learning. In *Proc. of ACL*, pages 1552–1561.
- Mitchell, M., Bohus, D., and Kamar, E. (2014). Crowd-sourcing language generation templates for dialogue systems. In *Proc. of INLG and SIGDIAL*, pages 24–32.
- Nenkova, A., Gravano, A., and Hirschberg, J. (2008). High frequency word entrainment in spoken dialogue. In *Proc. of ACL-HLT*, pages 169–172.
- Parent, G. and Eskenazi, M. (2010). Lexical entrainment of real users in the Let’s Go spoken dialog system. In *Proc. of Interspeech*, pages 3018–3021.
- Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskenazi, M. (2005). Let’s go public! taking a spoken dialog system to the real world. In *Proc. of Interspeech*.
- Reitter, D., Keller, F., and Moore, J. D. (2006). Computational modelling of structural priming in dialogue. In *Proc. of NAACL-HLT*, pages 121–124.
- Rudnicky, A. I., Thayer, E. H., Constantinides, P. C., Tchou, C., Shern, R., Lenzo, K. A., Xu, W., and Oh, A. (1999). Creating natural dialogs in the Carnegie Mellon Communicator system. In *Proc. of Eurospeech*.
- Serban, I. V., Lowe, R., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *arXiv:1512.05742*.
- Stoyanchev, S. and Stent, A. (2009). Lexical and syntactic priming and their impact in deployed spoken dialog systems. In *Proc. of NAACL-HLT*, pages 189–192.
- Vejman, M. (2015). *Development of an English public transport information dialogue system*. Master’s Thesis, Charles University in Prague.
- Wen, T.-H., Gasic, M., Kim, D., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015a). Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proc. of SIGDIAL*, pages 275–284.
- Wen, T.-H., Gasic, M., Mrkšić, N., Su, P.-H., Vandyke, D., and Young, S. (2015b). Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proc. EMNLP*, pages 1711–1721.
- Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., and Yu, K. (2010). The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

¹⁰Development continues on GitHub at https://github.com/UFAL-DSG/alex_context_nlg_dataset.

Dead Man Tweeting

David Nilsson[†], Magnus Sahlgren* and Jussi Karlgren*

[†]Nepa, Maria Skolgata 83, 118 53 Stockholm, Sweden

*Gavagai, Slussplan 9, 111 30 Stockholm, Sweden

david.nilsson@nepa.com, [firstname.lastname]@gavagai.se

Abstract

This paper presents a prototype — *Dead Man Tweeting* — of a system that learns *semantic avatars* from (dead) people’s texts, and makes the avatars come alive on Twitter. The system includes a language model for generating sequences of words, a topic model for ensuring that the sequences are topically coherent, and a semantic model that ensures the avatars can be productive and generate novel sequences. The avatars are connected to Twitter and are triggered by keywords that are significant for each particular avatar.

Keywords: Chat bot, language model, natural language generation

1. Introduction

Microblog services such as Twitter attract a significant amount of non-human users. It has recently been estimated that some 8% of all Twitter users are in fact bots.¹ Many of these are spambots or simple retweeters, but there are also examples of more sophisticated and creative bots.² This paper introduces a new type of Twitterbot: a *semantic avatar*, which is a reactive language model built from a person’s texts. The semantic avatar can react to input (in this case, tweets) that are of interest to it, and can stay updated with current events and new terminology.

The following sections describe our prototype implementation called *Dead Man Tweeting*, whose main purpose is to build semantic avatars of long-gone celebrities and to make them tweet. The current prototype features two semantic avatars: one for Russian author Fyodor Dostoyevsky (1821–1881), and one for British author and politician Winston Churchill (1874–1965).

2. System Architecture

The semantic avatars are composed of a combination of three modules: one module for generating text sequences, another module for controlling the topical structure of each sentence, and a third module that keeps track of semantically related terms. The three modules are initialized with some given text corpus (e.g. the collected works of a dead author), and the output of the last processing step (the semantic model) is connected to Twitter. Note that the semantic module also receives input from the output domain.

2.1. The Language Model

The first module is a sequence generator based on an n -gram language model, which gives a probability distribution for a sequence w_1, \dots, w_m as $\prod_{i=1}^m P(w_i | w_{i-(n-1)}, \dots, w_{i-1})$ given a context window of n words. As an example, an n -gram language model trained on a sample of general English language would give high probability to words such as “old” and “fast” when querying for the next word of the sequence “my car

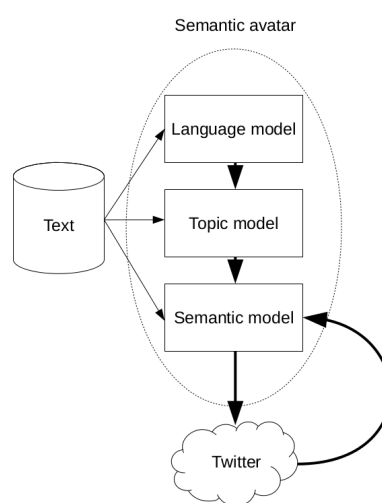


Figure 1: System architecture of Dead Man Tweeting

is very,” while words such as “the” and “yogurt” would get a low probability. This fits well with our intuitions about syntactic plausibility; you seldom hear anyone say “my car is very yogurt.”

The main problem with n -gram language models is that they do not generalize well, and are typically only able to reproduce sequences from the training data. More complex models such as Conditional Random Fields (CRF) (Roark et al., 2004), Recurrent Neural Networks (RNN) (Mikolov et al., 2010), and the currently very popular RNN variant Long-Short Term Memory (LSTM) (Wen et al., 2015) attempt to overcome these limitations, but they are often computationally intensive, and typically require considerable amounts of training data.

In the type of scenario we are concerned with here – building a language model based on some person’s texts – we typically have access to limited amounts of training data. Even the collected works of the most productive writers are minute in comparison with the size of current standard corpora such as the British National Corpus (100 million words),³ Wikipedia (some 1 billion words),⁴ or the WaCKy

¹qz.com/248063

²en.wikipedia.org/wiki/Twitterbot

³natcorp.ox.ac.uk

⁴dumps.wikimedia.org

corpora (more than 1 billion words).⁵

The scarcity of training data coupled with concerns of efficiency are the main reasons we opt for a simple n -gram language model with $n = 5$. We handle the limited capacity for generalization and productivity of the n -gram model by using two additional external modules: a topic model and a semantic model.

2.2. The Topic Model

The second module is a *topic model* that computes topical similarity between words. There are many ways to do this, ranging from simple methods such as the Vector Space Model (VSM) (Salton, 1964) to more advanced statistical models such as Latent Semantic Analysis (Landauer and Dumais, 1997), or probabilistic models such as Latent Dirichlet Allocation (Blei et al., 2003), and its derivatives. Again, since the more complex methods tend to be computationally intensive and require large amounts of training data, we opt in this application for the simple VSM, which represents each word as a vector $\vec{w} = [t_1, \dots, t_n]$ where t_i is the *tfidf*-weight of the word in the i th text region.⁶ Words that often occur in the same text regions get similar vectors in the VSM, and we can thus use the resulting vectors to control for topical coherence of the sequences generated by the n -gram model.

2.3. The Semantic Model

The semantic model is a *distributional semantic model* (Turney and Pantel, 2010), in which each word is represented by a *distributional vector*, $\vec{w}_f = [w_1, \dots, w_m]$ where w_i is a function of the co-occurrence count between the focus word w_f and each context word w_i that has occurred within a window of k tokens around the focus word. Words that have co-occurred with the same *other* words (i.e. that are interchangeable in context) get similar vectors in the semantic model, which means we can use the semantic model to substitute words suggested by the n -gram model.

To implement the distributional semantic model we use the Random Indexing framework (Kanerva et al., 2000; Sahlgren et al., 2008), which accumulates distributional vectors by summing sparse random *index vectors* of fixed size that act as fingerprints for each context word. Updating a simple “one-hot” distributional model would lead to a progressively increasing dimensionality with increasing vocabulary, but the Random Indexing model accommodates the entire vocabulary in a vector space of fixed size. In effect, Random Indexing can be seen as an incremental random projection of a distributional semantic model. In this prototype, we use 2,000-dimensional vectors and 10 non-zero elements in the random index vectors. The context window is set to $k = 4$.

⁵wacky.sslmit.unibo.it

⁶ $tfidf_{i,j} = tf_{i,j} \cdot \log \frac{N}{df_i}$ where $tf_{i,j}$ is the frequency of word i in document j , df_i is the number of documents word i has occurred in, and N is the total number of documents in the data. A “document” is often a paragraph in the type of data used in these experiments.

3. The Semantic Avatars

The three modules described above are initialized with all available texts for the person in question. We have selected two prominent authors as test pilots for the semantic avatars: Fyodor Dostoyevsky, one of the authors to define the modern novel, and Winston Churchill, the 1953 Nobel Prize Laureate in Literature. Both of these authors were selected because there are substantial amounts of data freely available on sites such as Project Gutenberg.⁷ For Dostoyevsky, the downloaded data contains some 1 million words, while for Churchill, the data consists of approximately 1.6 million words.

The initialization phase produces an n -gram model, a topic model, and a semantic model for each author. The n -gram model is able to generate sequences that are coherent at a very local level, but may be incoherent over longer sequences. This is where the topic model comes in. While the n -gram model learns about local structures, the topic model learns about global structures in the sense that it can recognize topically related words. Thus, if we sample the succeeding word given $n - 1$ previous words using the n -gram model, we can use the topic model to filter out words that are not topically related to the words that are already in the current sequence. This topic filter ensures that each sequence is topically coherent even over longer sequences. However, we are still only able to reproduce short sequences of words we have already seen in the original data, and since our chosen subjects have not generated any words for a long time, their avatars have quite outdated vocabulary. Enter the semantic model, which is partly trained on the subject’s texts, but which also receives an input stream from relevant news channels. In the case of Dostoyevsky, we use feeds from Reuters and Russia Today, and in the case of Churchill we use a feed from BBC news. These input streams are continuously fed into the semantic model, which thereby accrues an expanding vocabulary that may contain terms that were never used by the avatar’s ancestor. The semantic model thus constitutes an avatar’s ability to learn new words that are in some sense related to what it already knows. The avatars may use the semantic model to randomly substitute words generated by the n -gram model to other words that are highly related in the semantic model. The effect is that the avatars are able to use terminology that was not present in the original data, which means they can now be productive and generate completely novel sequences.

The combination of the three different modules enables the semantic avatars to not only generate topically coherent sequences, but also to be productive and utter new things. However, we also want the avatars to be reactive and to interact with the world. We therefore connect each avatar to Twitter and add some accounts to follow. Each avatar is triggered by tweets containing words of particular interest to the avatar, which are extracted from the training data using *tfidf*. If the avatar is triggered by a tweet containing a keyword, it formulates a reply containing that keyword, and posts it on Twitter.

⁷gutenberg.org

4. Examples

Figures 2 and 3 give examples of how Churchill’s semantic avatar (@churchillDMT) and Dostoyevsky’s semantic avatar (@dostoyevskyDMT) interact with other Twitter users; in the first example, Churchill’s semantic avatar refutes a claim made by the Churchill Center, and in the second example, Dostoyevsky’s semantic avatar comments on a tweet by the Russian Ministry of Foreign Affairs. The two avatars sometimes also comment on each other’s tweets, as in Figure 4 when Churchill’s semantic avatar comments on a tweet by Dostoyevsky’s avatar (which in its turn is a reply to another tweet). There are also examples of when other Twitter users reply to the avatar’s tweets, apparently considering them (correctly so) to be communicative users, as in Figure 5.

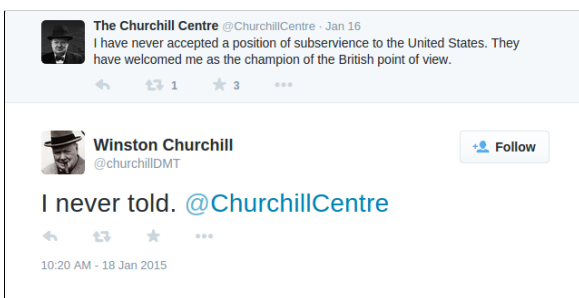


Figure 2: Churchill’s semantic avatar answers the Churchill Center.



Figure 3: Dostoyevsky’s semantic avatar answers the Russian Ministry of Foreign Affairs.

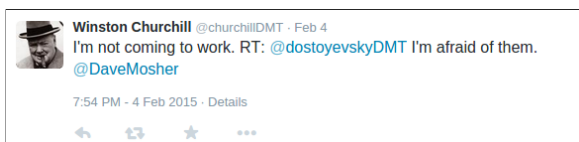


Figure 4: Churchill’s and Dostoyevsky’s semantic avatars having a conversation.

5. Discussion

The examples above are of course hand-picked to demonstrate when the avatars produce intelligible (or at least entertaining) conversation. Admittedly, the avatars do not always produce meaningful output. Two telling examples are @dostoyevskyDMT’s tweet “But she after three

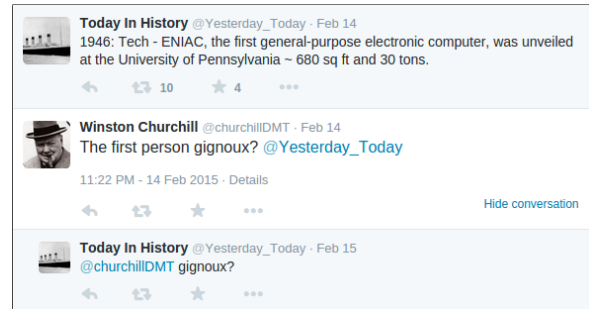


Figure 5: Churchill’s semantic avatar converses Today In History, but makes little sense.

years later!” and @churchillDMT’s tweet “But most men in the working women relief.”

There are many reasons for the varied quality of the avatar’s responses. First of all, there are a number of parameters in the interaction between the various modules that can be tuned and optimized; when optimizing for short sequences, the avatars produce very terse responses, while when optimizing for longer sequences, the avatars risk becoming incoherent. Furthermore, there is currently no topic correction between an avatar’s generated text and the tweet it reacted upon. We are at this stage merely requiring the presence of the trigger keyword, which is obviously a too lenient requirement that can cause incoherent responses. A component which would hold to a topic-comment structure, with a topic selected from the input stimulus but with a freer progression towards associated topics would give some level of discourse backbone to the text.

Also, the semantic model may cause alterations that make the sequence less intelligible, which could potentially be solved by letting the topic model override semantic expansions in order to preserve topical coherence.

It is of course well-known that language generation for open domain text is a very difficult task. Most previous studies and applications of natural language generation has focused on closed domains, such as weather forecasts (Belz, 2008), health care (Harris, 2008), or review summarization (Di Fabbrizio et al., 2013), and commercial vendors such as Automated Insights⁸, Narrative Science⁹, and Yseop¹⁰, also tend to focus on domain-specific solutions. One of the reasons for choosing Twitter as the medium for the semantic avatars rather than, e.g., email or blogs is that the length restriction of tweets makes it easier to generate open domain text and that the rhapsodic nature of the generated discourse better fits expected human behaviour. However, the strength of the Twitter medium – its brevity – is also a weakness in the current application, since it is difficult to replicate an author’s stylistic and topical peculiarities, and to demonstrate semantic productivity, in such short sequences. As a consequence, many of the avatars’ tweets take the form of questions, exclamations, or otherwise quite terse statements, where it is difficult to discern any personality traits.

⁸automatedinsights.com

⁹narrativescience.com

¹⁰yseop.com

Even so, the dialogic nature of Twitter facilitates a benevolent interpretation of the avatar’s utterances. Even very short and repetitive tweets, such as @dostoyevskyDMT’s “yes yes yes” may be interpreted as a perfectly viable reply given the context “Apple’s App Store says no to guns but yes to weed” tweeted by @TechCrunch. A Turing test based on Twitter would probably be passable using technologies such as those presented here, which would most likely not be the case given other communication channels.

We of course make no claims about any communicative intelligence behind the semantic avatars; they simply put words together – but they do put words together under much the same type of constraints as is characteristic of human communicative competence: syntactic (which we simplify to sequential) consistency, topical coherence, and semantic productivity. At the current state of the semantic avatars, the sequences they generate make sense on Twitter about as often as they seem to be gibberish. More data would likely improve the quality of the avatars.

6. Texts and Resources

We have implemented the semantic avatars not to need any other resources than text from the avatar’s living correlates: no need for grammars, hand-encoded knowledge structures, or other expensive conceptual models. Access to a sizeable text collection is thus a necessary prerequisite for launching an avatar. The model, as implemented in these present experiments, relies on both the collection of background text to give the avatars their personal touch, as well as access to text on current events to allow them to evolve with current language usage, and to provide a general language background. Collections of general language background data from various historical periods and covering various types of domains and styles would be a very useful resource for the creation of historic avatars.

The various modules of the avatars can be seen as individual resources for further development of personalized chat bots. Harvesting sequence, topical, and semantic modules from a large number of subjects would enable the creation of *modular* avatars, where it would be possible to select a sequence model from one subject, a topical model from another, and a semantic model from a third. The resulting combined avatar would have the syntactic qualities from one person, the topical knowledge of another, and the semantics of a third; imagine a bot that would have the syntactic qualities of James Joyce, the topical knowledge of Aristotle, and the semantics of Nietzsche.

7. Directions of Application

We have used the prototype to revive two prominent dead authors, and one could imagine applications geared towards *mind uploading*¹¹ – i.e. the postmortem preservation of personality – which would enable people to converse with the dead, and to solicit opinions and statements on current events. A bit less science fiction-like, one might also use this type of system for creating communicative avatars for

entertainment purposes (e.g. chat bots, or in-game characters), or for creating aggregated views on entire text collections, such as a political party’s or a company’s internal documents. Currently we are including a Dead Man Tweeting component as a fall-back resource in a chat bot geared towards school children for a specific field of discussion. When the conversation veers off target or leaves the fairly limited area of competence the bot has been equipped with (which can be expected when conversing with teenagers), the chat bot uses a Dead Man Tweeting component to generate a fall-back response. In this case, the text corpus used for training is a parameter to be experimented with to get the most natural and entertaining output.

8. Conclusion

This paper has introduced the notion of a semantic avatar, and has provided examples of output from two different avatars: @dostoyevskyDMT and @churchillDMT. The avatars are based on a combination of a language model that generates sequences of words, a topic model that ensures the generated sequences are topically coherent, and a semantic model that equips the avatars with semantic productivity.

Despite leaving room for improvement, we find that the examples produced by the Dead Man Tweeting prototype demonstrate the potential for semantic avatars as a way to build autonomous reactive systems that can interact linguistically with its surroundings.

9. References

- Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, October.
- Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Di Fabbri, G., Stent, A., and Gaizauskas, R. (2013). Summarizing opinion-related information for mobile devices. In Amy Neustein et al., editors, *Mobile Speech and Advanced Natural Language Solutions*, pages 289–317. Springer New York.
- Harris, M. D. (2008). Building a large-scale commercial NLG system for an EMR. In *Proceedings of INLG*, pages 157–160.
- Kanerva, P., Kristofersson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of CogSci*, page 1036.
- Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048.
- Roark, B., Saraclar, M., Collins, M., and Johnson, M. (2004). Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of ACL*.

¹¹en.wikipedia.org/wiki/Mind_uploading

- Sahlgren, M., Holst, A., and Kanerva, P. (2008). Permutations as a means to encode order in word space. In *Proceedings of CogSci*, pages 1300–1305.
- Salton, G. (1964). A flexible automatic system for the organization, storage, and retrieval of language data (SMART). In *Report ISR-5 to NSF*, Cambridge, MA. Harvard Computation Laboratory.
- Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, January.
- Wen, T.-H., Gasic, M., Mrksic, N., Hao Su, P., Vandyke, D., and Young, S. J. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of EMNLP*, pages 1711–1721.

Chatbot Evaluation and Database Expansion via Crowdsourcing

Zhou Yu, Ziyu Xu, Alan W Black, Alexander I. Rudnicky

Carnegie Mellon University

5000 Forbes Ave, Pittsburgh, 15217

{zhouyu, air, awb}@cs.cmu.edu, ziyux@andrew.cmu.edu

Abstract

Chatbots use a database of responses often culled from a corpus of text generated for a different purpose, for example film scripts or interviews. One consequence of this approach is a mismatch between the data and the inputs generated by participants. We describe an approach that while starting from an existing corpus (of interviews) makes use of crowdsourced data to augment the response database, focusing on responses that people judge as inappropriate. The long term goal is to create a data set of more appropriate chat responses; the short term consequence appears to be the identification and replacement of particularly inappropriate responses. We found the version with the expanded database was rated significantly better in terms of the response level appropriateness and the overall ability to engage users. We also describe strategies we developed that target certain breakdowns discovered during data collection. Both the source code of the chatbot, TickTock, and the data collected are publicly available.

Keywords: crowdsourcing, dialog systems, appropriateness

1. Introduction

Chatbots have recently become the focus of greater research interest. Unlike goal oriented dialog systems, chatbots do not have any specific goal that guides the interaction. Consequently, traditional evaluation metrics, such as task completion rate, are no longer appropriate. The difficulty of evaluation is intrinsic as each conversation is interactive, and the same conversation will not occur more than once; one slightly different answer will lead to a completely different conversation; moreover there is no clear sense of when such a conversation is “complete”. It is not possible to design a pipeline to evaluate such systems in a batch mode, nor is it easy to equate participants on various dimensions that may influence their behavior.

In addition to the difficulty of evaluating a chatbot, another challenge is identifying an appropriate database. Ideally, it should be a corpus that has the same distribution as the future users’ conversations. However, if we are not designing a specific system for a targeted group, but rather a system that can be used by a variety of different users, the best strategy for designing a response database is not clear.

We describe an approach we have developed to the problem of creating a database of useful responses that makes use of an existing corpus as a base but also incorporates a process that iteratively transforms the database into a form that is better suited to the chat domain. We do this by crowdsourcing the appropriateness of responses (in given conversations) and by soliciting improved responses.

2. Related Work

Current chatbots use a variety of methods to generate responses, such as machine translation (Ritter et al., 2011), retrieval based response selection (Banchs and Li, 2012), and recurrent neural network sequence generation (Vinyals and Le, 2015). Yet, the databases they use to power their systems have very little variability. Some systems used micro-blogs, such as Twitter conversations (Ritter et al., 2011) and some used movie subtitles (Banchs and Kim, 2014; Ameixa et al., 2014; Banchs and Li, 2012), and there is research that used Twitter as a database but switched to

ask the human to generate responses in the crowdsourcing platform in real time when the database failed to have an appropriate response (Bessho et al., 2012). Most of the work reported above have no real user evaluation or a small group of people for evaluation. Only two kinds of databases have been used, movie subtitles and micro-blogs. In this work, we focus on how to generate appropriate databases for chatbots and conduct evaluations for chatbots by leveraging crowdsourcing resources.

3. TickTock System Description

TickTock is a system that is capable of conducting free-form conversations, in contrast to goal-driven systems, which are designed to acquire information, provide feedback, or negotiate constraints with the human. A free-conversation system in principle removes any built-in value for the human and its success depends on the machine keeping the human interested in the ongoing conversation. Thus, as task completion is no longer an applicable metric, we chose to focus on metrics of the user’s experience, such as engagement, likability, and willingness to future interaction. TickTock is an IR-based system with conversation strategy facilitation. A multimodal version of TickTock is described in (Yu et al., 2015), with similar architecture but with minor adjustments to the conversational strategies.

TickTock 1.0 has a database that consists of question-answer pairs from CNN Interview Transcripts from the “Piers Morgan Tonight” show. The corpus has 767 Interviews in total and each interview is between 500 to 1,000 sentences. To construct our database, we used a rule-based question identification method, which simply means searching for tokens such as ‘?’, ‘How’, ‘Wh-’, etc. to identify questions and then extracted the consecutive utterance of the other speaker as the answer to that question. In total we have 67,834 pairs of utterances.

Key Term Matching (Martin, 2002) was used for response generation. The user’s text input is first processed by the NLU component, which performs POS tagging (Toutanova et al., 2003) and removes stop words; heuristics are then used to compute the database and calculate the weighted

sum, which becomes the retrieval confidence score. Finally, we normalize the score by dividing it by the length of the retrieved utterance. We filter out inappropriate content, excluding the retrieved answer if it is longer than 15 words and remove other characters such as parentheses or square brackets (along with everything between them). Our goal is to generate coherent conversations without deep understanding of the context, which is useful in a non-task oriented interactive system, and is motivated by lexical cohesion in modeling discourse. The coherence can be reflected by the repetition of lexicon items. The method first does shallow syntactic analysis of the input utterance and extracts keywords. These are used to search the corpus for a suitable response. Once we retrieved the response, we select a conversational strategy, based on a heuristic, i.e. a predefined threshold for the retrieval confidence score, which can be tuned to make the system appear more active or more passive.

Higher thresholds correspond to more active user engagement. When the retrieval confidence score is high, we return the found response in the database back to the user. If the retrieval confidence score is low, meaning no good response was obtained, we use strategies to change the current topic by randomly choosing four types of conversation strategies we designed. “Proposing a new topic”, such as “sports” or “music”; “Closing the current topic using an open question, such as “Could you tell me something interesting?”; “Telling a joke”, such as “ Politicians and diapers have one thing in common. They should both be changed regularly, and for the same reason”; and finally “Initiate things to do together”, such as “ Do you want to play a game together?”

4. Methodology

The purpose of the crowdsourcing study is twofold. The first is to collect diverse conversations from a large number of people. The second is to expand TickTock’s database, so it has more targeted question-response pairs. The TickTock system is implemented in Python, making it platform independent. We adapted TickTock to a web version through a web socket connection to a web page implemented in PHP. People can get access to TickTock through any browser. We made the source code of TickTock, a web-based demo and a collection of 100 conversations on Mechanical Turk publicly available here¹.

We designed three crowdsourcing tasks to expand TickTock’s database. The first task is “the conversation generation task”, in which a user interacts with TickTock by typing. The second task is “the conversation rating task”, in which the user rates how appropriate TickTock’s response is per conversational turn. The third task is “the conversation correction task”, in which the user generates appropriate responses for TickTock. For the last task, we only selected the conversational turns that were rated not appropriate in the second task for correction by Turkers.

We recruited participants on the Amazon Mechanical Turk Platform with Turker criteria of: higher than 95% life time

approval rate, completed more than 50 hits, and located in the United States.

After we had collected sufficient data from the above three tasks, we expanded our database by adding the human approved high-appropriateness responses obtained from the second task and the human corrected responses obtained from the third task to TickTock’s database. The system with the expanded database is named TickTock 2.0. The new version was then put on the Amazon Mechanical Turk Platform to collect more data. After we collected more conversations and the corresponding appropriateness ratings for those conversations, we calculated the appropriateness rating distribution of the two versions of TickTock to determine if there was improvement after we expanded the database with more targeted data.

We also collected user subjective ratings for the two versions. We asked users how engaged they felt over all in their interactions. Users who interacted with both versions of the system were also asked whether they preferred TickTock 2.0.

5. Mechanical Turk Study Designs

We designed three tasks and recruited participants on the Amazon Mechanical Turk Platform. We describe the task design and the collected data in this section.

5.1. Conversation Generation Task

We asked the user to interact with TickTock via the web page for at least 10 conversational turns. We also collected information from users on whether they had interacted with a chatbot before. Users were also asked to state what they liked, and disliked about the task; this was for purposes of providing insight for system improvement. The task starts when user types in an utterance on a web page, and submit it to the server, the server then fetches a response and displays it to the user. For both versions, we collected 50 conversations. For TickTock 1.0, this took over four days, with 44 participating Turkers (we allow people to do the same task multiple times), resulting in 589 conversation turns. The conversation length on average was 11.9, with a standard deviation of 1.9. With TickTock 2.0, it took over ten days, with 29 Turkers, and resulted in 590 turns. The conversation length on average was 11.8, with a standard deviation of 4.2. We are not certain why the two versions took different amounts of time; perhaps our task was of limited overall interest.

5.2. Conversation Rating Task

Once the Turker finished his or her conversation with TickTock, we asked them to rate how appropriate they felt the system’s responses were with respect to their inputs. We also told them to make the decision for each conversational turn. Table 1 describes the annotation scheme that Turkers were given. The Turker would see the entire dialog on the web page, and were asked to choose between three labels: ‘Inappropriate’, ‘Interpretable’ and ‘Appropriate’.

We randomly sampled 10 percent of the collected utterance pairs and asked an expert to rate how appropriate TickTock’s responses are given the same coding manual Table 1. Since we wanted to collect conversational turns

¹<http://www.cs.cmu.edu/afs/cs/user/zhouyu/www/TickTock.html>

Label	Definition	Example
Inappropriate	Not coherent with user utterance	<i>Participant</i> : How old are you? <i>TickTock</i> : Apple.
Interpretable	Related and can be interpreted	<i>Participant</i> : How old are you? <i>TickTock</i> : That’s too big a question for me to answer.
Appropriate	Coherent with the user utterance	<i>Participant</i> : How is the weather today? <i>TickTock</i> : Very good.

Table 1: Appropriateness rating scheme.

that are not appropriate and send them back to Turkers to generate more appropriate responses for TickTock, we collapsed ‘Inappropriate’ and ‘Interpretable’ into ‘Not Appropriate’ when doing the annotation agreement as we wanted to distinguish between ‘Not Appropriate’ and ‘Appropriate’. The agreement of the participant’s self-rated appropriateness and the expert-rated appropriateness has a kappa of 0.73. In Table 2, we display an example dialog with its associated appropriateness ratings.

5.3. Conversation Correction Task

Turkers were shown three utterances: one utterance from the participant, one utterance from TickTock and another utterance from the participant. Then, they were asked to type in what they should say if they were TickTock given the three utterances. The original TickTock’s response to the previous utterance from the participant was not shown. In total, 28 Turkers participated in this task.

We randomly sampled 10 percent of the corrected conversational turns and asked an expert to rate how appropriate the responses were, according to the appropriateness rating scheme mentioned above. We found that 82.8% of the responses were appropriate, and the inappropriate responses were just answers with different lengths of the letter ‘d’, which is what people put to get through the task in the most efficient way. We filtered these bad responses out based on a simple regular expression. The appropriate percentage thus increased to 100%. We conjecture that the reason this task appears very easy for Turkers might be their experience in conducting conversations with others.

6. Results and Analysis

We would judge that our approach is a reasonable way to generate focused chat data: We spent approximately \$50 for the experiments we conducted on the Amazon Mechanical Turk Platform and collected 50 conversations for each of the two versions of the system. The experiments lasted 14 days in total. As one would expect, we found it usually takes less time to complete the task if you raise the payment of the task and is therefore a decision that researchers can make according to their priorities. In Table 3, we show the distribution of the rating of two systems’ response appropriateness. The inappropriateness ratio of turns has dropped from 55% to 34% by expanding the database with appropriate question-answer pairs.

There are in total eight people who have interacted with both versions of TickTock, in which five of them preferred the second version, two of them think the two versions are

the same, and the remaining one was not sure on their preference. The average user self-reported engagement score is 2.4 (out of 5) in the TickTock 1.0 experiment and 3.6 (out of 5) in TickTock 2.0. We observe that TickTock 2.0 is performing better than TickTock version 1.0 in both per turn appropriateness measure, and per interaction user engagement measure. There are only around 10 percent of users (8/(44+29)) who interacted with both versions of the system. In the future, we would like to assemble a pool of Turkers that we can continue asking to interact with different versions of the chatbot, for evaluation consistency.

After filtering stop words, we did a word count of all the utterances users typed in and found that none of the top 50 ranked words were content words. Most of them were functional words, such as ‘you’, ‘I’, ‘is’, ‘do’, ‘are’, ‘what’, similar to what has been observed in human conversation.

We also performed a POS tagging and extracted the nouns used by participants; we found the top ranked words to be ‘music’, ‘movie’, ‘game’, ‘sense’, ‘name’, ‘TickTock’, and ‘Trump’. We configured TickTock to initiate conversation about everyday topics such as music and movies when lacking an appropriate answer, which may have led participants to expand on those topics in conversation. Some participants got annoyed and said ‘You don’t make sense’ or ‘It doesn’t make sense’, causing ‘sense’ to be at the top of the list as well. We calculated the percentage of pronouns used in user sentences. The percentage, 9.4%, is relatively high. This is another phenomenon that seems to be related to the word choice in casual conversation.

We also went through a randomly selected 10% of the data, and observed that the quality of the users’ input is not always very satisfying. Most of the users failed to provide concrete inputs to the conversation. They asked Ticktock simple questions or simply went along with whatever TickTock was saying instead of pushing the conversation topic deeper. How to elicit high quality user input is a difficult question; very few researchers in the chatbot community appear to have paid attention to it. We would argue that not only is an appropriateness metric for TickTock’s response needed, but also an additional metric that evaluates the quality of a user’s input. Together they may better characterize chatbot performance.

7. Strategies Designed for Different Types of Breakdowns

We randomly sampled 10% of the conversations generated by using TickTock 1.0 and classified the nature of conversational breakdowns (turns given ratings that are not ‘Appro-

Participant	TickTock(Appropriateness)
Hello how are you?	I'm feeling quite good. (3)
Anything new going on with you?	Of course. What do you think? (3)
I am not sure. What's on your mind?	Well, you know, I'm not sure. let me tell you something, Politicians and diapers have one thing in common. They should both be changed regularly, and for the same reason. (3)
Wow. That funny and insightful.	Uhh, I can't answer this. Can we talk about something else? (2)
How about movies. Have you seen any good movies?	Of course. What do you think? (2)
What have you seen?	Yes. (2)
I love super hero movies. Have you seen Deadpool?	Uh-huh, I do. (2)

Table 2: An example conversation with TickTock

Version	Inappropriate	Interpretable	Appropriate
1	321 (55%)	138 (23%)	130 (22%)
2	200 (34%)	242 (41%)	148 (25%)

Table 3: The distribution of appropriateness ratings of two versions of TickTock.

appropriate') into five types. We formulated targeted strategies for each type and evaluated them on the data collected by TickTock 2.0.

1. **Single-word Sentence:** We found that some users were typing in meaningless single words such as 'd', 'dd', or equations such as '1+2='. TickTock will reply 'Can you be serious and say things in a complete sentence?'. We have a set of surface realization of such replies to choose from, so users would get a lightly different version every time, with the aim of making TickTock seem less robotic. It triggered 12 times in the TickTock 2.0 generated conversations.
2. **Out of Vocabulary:** We found that typos occur in the users' responses and they used words that are not in the vocabulary of our database, such as 'confrontational'. We implemented a strategy that when a sentence contains an out of vocabulary word, TickTock will reply with a clarification question, such as 'What is 'confrontational'?' to communicate that it cannot understand his utterance entirely. It triggered 36 times in the TickTock 2.0 generated conversations.
3. **Anaphora:** We found user inputs with very limited concrete information in themselves, but referred to a prior response in the conversation. An example input would be "I hate them" and it is referring back to the 'sports' topic in the previous phrase, "How about we talk about sports?". Anaphora is a difficult problem to solve for complex sentence structures. However in colloquial sentences, substituting in the noun of the previous sentence covers 85% of the cases. We implemented this simple rule to tackle anaphora. It triggered 30 times in the TickTock 2.0 generated conversations.
4. **Query Knowledge Base for Named Entities** A lot of Turkers assumed TickTock could answer factual questions, so they asked questions such as "Which state is Chicago in?". We used the Wikipedia knowledge base API to answer such questions. We first performed a

shallow parsing to find the named entity in the sentence, which we then searched for in the knowledge base, and retrieved the corresponding short description of that named entity. We then designed a template to generate sentences using the obtained short description of the mentioned name entity, such as "Are you talking about the city in Illinois?". It triggered 22 times in the TickTock 2.0 generated conversations.

5. **Weight adjustment with tf-idf** We re-weighted the importance of the key words in an utterance based on its tf-idf score. Using POS tagging of the words that match between a user input, and the sentence a response is in reply to, we give nouns a score of 3, verbs a score of 2, and other words a score of 1. We then multiply each of these scores by the tf-idf value of the corresponding words, and the sum of their scores gives us the score of the response.

8. Conclusions and Future Work

We found that using suitable designed crowdsourcing tasks, we can expand TickTock's database with more targeted response pairs. The version using the expanded database was preferred by most of the users and was better rated in terms of response appropriateness and the overall ability to engage users. We also found it is feasible to use the crowdsourcing platform for system evaluation. An analysis of the data we obtained also allowed us to define strategies to recover from breakdowns (some of which have previously been reported by others).

Our intent is to go beyond the response appropriateness and put more emphasis on overall discourse cohesion. For example, there is a breakdown type we have not addressed, which is the chatbot's inconsistency in adhering to the context of the conversation. A possible solution would be to maintain a knowledge base of what the user said and use it for consistency checking as part of the selection process for the final response.

We are also interested in determining how the system can channel a conversation into a specific topic. That is, if TickTock starts the conversation with a given topic, how long and with what strategies will it be able to keep the user on the same topic. We also wish to develop strategies that elicit high quality responses from human users (perhaps as a consequence of maintaining a high level of engagement).

9. Acknowledgements

We thank Peter Xiang Gao for technical supports and discussions.

10. References

- Ameixa, D., Coheur, L., Fialho, P., and Quaresma, P. (2014). Luke, i am your father: dealing with out-of-domain requests by using movies subtitles. In *Intelligent Virtual Agents*, pages 13–21. Springer.
- Banchs, R. E. and Kim, S. (2014). An empirical evaluation of an ir-based strategy for chat-oriented dialogue systems. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (AP-SIPA)*, pages 1–4. IEEE.
- Banchs, R. E. and Li, H. (2012). Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the ACL 2012 System Demonstrations*, pages 37–42. Association for Computational Linguistics.
- Bessho, F., Harada, T., and Kuniyoshi, Y. (2012). Dialog system using real-time crowdsourcing and twitter large-scale corpus. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 227–231. Association for Computational Linguistics.
- Martin, J. R. (2002). *Meaning beyond the clause: area: self-perspectives*. *Annual Review of Applied Linguistics* 22.
- Ritter, A., Cherry, C., and Dolan, W. B. (2011). Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.
- Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.
- Vinyals, O. and Le, Q. (2015). A neural conversational model. *ICML Deep Learning Workshop 2015*.
- Yu, Z., Papangelis, A., and Rudnicky, A. (2015). Tick-Tock: A non-goal-oriented multimodal dialog system with engagement awareness. In *Proceedings of the AAAI Spring Symposium*.

Framework for the Formulation of Metrics for Conversational Agent Evaluation

Mohammed Kaleem, Omar Alobadi, James O'Shea, Keeley Crockett

Intelligent Systems Research Group
Department of Computing, Mathematics and Digital Technology
Manchester Metropolitan University
Manchester, United Kingdom

E-mail: (m.kaleem, j.d.oshea, k.crockett)@mmu.ac.uk, oalobadi@yahoo.com

Abstract

The evaluation of conversational agents is an area that has not seen much progress since the initial developments during the early 90's. The initial challenge faced when evaluating conversational agents is trying to formulate the metrics that need to be captured and measured in order to gauge the success of a particular conversational agent. Although frameworks exist they overlook the individual objectives of modern conversational agents which are much more than just question answering systems. This paper presents a new framework that has been utilised to formulate metrics to evaluate two conversational agents deployed in two significantly different contexts.

Keywords: Conversational agents, dialog agents, evaluation, evaluation metrics

1 Introduction

This paper illustrates the application of a software quality model in Conversational Agent (CA) evaluation. According to the IEEE Glossary of Software System Engineering Terminology (IEEE, 2000), quality is defined as the degree to which a system, a component, or a process meets customer or user needs or expectations. Roy and Graham (2008), posit that the quality of software is measured primarily against the degree to which requirements, such as correctness, reliability and usability are met. The factors that affect quality are termed as quality attributes or metrics. There are different categorizations of quality metrics. Roy and Graham (2008), further state that quality metrics can be categorized into two broad groups: metrics that can be directly measured (e.g. performance) and metrics that can be indirectly measured (e.g. usability). These metrics can be translated into objective and subjective metrics respectively. In order to build a successful dialogue system, data is needed on how users behave and their perceptions when interacting with the system (Skantze and Hjalmarsson, 2013). Recent work in the field has produced CAs in very diverse applications (Keeling et al., 2004; Alobaidi et al., 2013; Latham et al., 2014), therefore the evaluation of such systems has to suit the individual goals of the application domain. As such the evaluation metrics cannot be generalized in to a "one size fits all" evaluation framework. The weakness with existing frameworks is that they fail to identify the individual evaluation metrics that need to be gauged, metrics that are unique to the goal of the CA developed.

This paper is structured as follows: Section 2 outlines the concept of CAs and presents the existing work conducted in the field of CA evaluation. Section 3 details the proposed framework for formulating CA evaluation metrics and details two case studies where the framework was used to evaluate two different CAs. Section 4 discusses the results of the case studies. Section 5 details the conclusion drawn from the case studies.

2 Background

2.1 Conversation Agents

The term "Conversational Agent" (CA) is interpreted in different ways by different researchers; however the

essence of CAs is natural language dialogue between the human and an application running on a computer (O'Shea et al., 2011). Recent developments in the field of CAs have utilized complex artificial intelligence techniques in order to facilitate a rich goal driven conversation with the user. These types of CAs have been applied in a wide array of contexts such as a CA Help Desk: responding to employee or customer questions related to complex processes or procedures (Lester et al., 2004; Kaleem et al., 2014), Website Navigation/Concierge: guiding customers to relevant portions of complex websites (Shimazu, 2002), Guided Selling: providing answers and guidance in the sales process, particularly for complex products being sold to novice customers (Keeling et al., 2004), Education: known as Conversational Intelligent Tutoring Systems (CITS) (Alobaidi et al., 2013; Latham et al., 2014) and HR Bully and Harassment Help System (Latham et al., 2010).

2.2 Conversation Agent Evaluation

According to Martinez et al. (2008), it is quite difficult to evaluate dialogue systems. In addition to the lack of evaluation standards within the dialogue community, it is difficult to find performance figures from real world applications that can be extrapolated to other systems or be accepted worldwide, as all of them are directly related to one specific dialogue system. Although CA/Dialogue system evaluation frameworks exist, these frameworks are dated and moreover they generalize the metrics tested between individual systems, therefore overlooking the increasingly complex developments in the field of CAs. CAs are now more than just question answer systems.

An early example of evaluating the success of dialog based software is the Turing test. The Turing test (Turing, 1950) was primarily aimed at making a human believe that they were speaking to another human, when in fact they were speaking to a computer program. This approach however is not suitable to gauge the effectiveness or usability of a modern goal orientated conversational agent as the intrinsic nature behind the two applications are completely different.

Existing CA evaluation frameworks such as PARADISE were devised almost 20 years ago and while they were suited to evaluate the CAs of their time they are not entirely suitable to evaluate modern day conversational agents which are much more technologically advanced,

taking advantage of artificial intelligence to achieve much more diverse goals like tutoring (Latham et al., 2014) and offering specialist advice (Latham et al., 2010).

A substantial amount of work has been done on evaluating CAs as a whole. The seminal work in this area was done by Walker et al. (1997) who created the PARADISE framework which is a general framework for evaluating spoken dialogue systems. For determining the quality of Spoken Dialogue Systems, several aspects are of interest. Moller et al. (2009), presented a taxonomy of quality criteria. They describe quality as two separate issues consisting of Quality of Service and Quality of Experience. Quality of Service describes objective criteria like dialogue duration or number of turns or utterances it takes to achieve the desired outcome. While these are well-defined items that can be determined easily, Quality of Experience, which describes the user experience with subjective criteria, is a more vague area and without a sound definition, e.g. User Satisfaction.

There is a general agreement on “usability” as the most important performance figure in CAs (Turunen et al., 2006) even more than others widely used like “naturalness” or “flexibility”. However functionality may be more important, but without usability the system will not get the chance to demonstrate functionality. Therefore, besides quality and efficiency metrics, automatically logged or computed, subjective tests must also be performed in order to assess the impact of the capabilities of the system on user satisfaction and to get a valuable insight on the shortcomings and advantages of the system (Martinez et al., 2008). According to Silvervarg and Jönsson (2011), the evaluation of CA/dialogue systems is mainly done either by distributing a questionnaire to the users trying to reveal their subjective assessment of using the dialogue system or by studying the resulting dialogue. Artstein et al. (2009), refer to this as “soft” numbers versus “hard” numbers and propose a “semi-formal” evaluation method combining the two evaluation methodologies. This notion is supported by more recent research conducted by Rauschenberger et al. (2013) who propose a framework to measure user experience and software quality in interactive software applications through User Evaluation Questionnaires (UEQ). They state that the evaluation of interactive software quality falls into two distinct categories, these being “pragmatic quality” and “hedonic quality”. Pragmatic quality relates to task orientated quality like task completion effectiveness and efficiency. Hedonic quality is related to non-task orientated aspects like aesthetic impressions and user stimulation. These two categories can be translated into objective measures and subjective measures respectively.

Subjective aspects like user satisfaction are usually determined by using questionnaires with Likert Scale questions (Hone and Graham, 2000; Silvervarg and Jönsson, 2011; Rauschenberger et al., 2013). Objective metrics can be measured through records and logs of the user’s dialogue with the CA. These metrics are captured whilst a user is undergoing an evaluation session to achieve a pre-set task. The records/logs are used to capture and store several variables related to the dialogue such as rule fired, similarity strength, user utterance, CA response etc. Based on these captured variables which are stored in the log file, the CA can be evaluated for effectiveness accuracy and robustness, through statistical analysis. The general consensus among researchers in the field from the early

days to the present day is that the effectiveness of a CA/Dialogue system should be evaluated through a combination of subjective and objective measures (Alobaidi et al., 2013; O’Shea et al., 2011; Rauschenberger et al., 2013; Walker et al., 1997). This ensures that not only is the effectiveness of the CA’s functionality tested but the usability from the user perspective is also tested. It has been established that there is standard set of metrics related to usability and task completion that are to some extent universal for the evaluation for all dialogue systems.

However there is no framework which can be followed to derive the individual metrics that need to be tested in order to evaluate the success of modern day conversational agents that are much more than just general chat applications - they are tutoring users on diverse topics and explaining complex procedures.

As there has been no formal development of the CA evaluation frameworks over the years, alternative existing approaches/evaluation frameworks that can be adopted are software evaluation frameworks that that are utilized to test new software applications in terms of functionality and usability (i.e. objective and subjective metrics).

3 Novel Framework for the Formulation of Evaluation Metrics Suited to Individual CA Goals

As with any engineering discipline, software development requires a measurement mechanism for feedback and evaluation. Measurement is an aid in answering a variety of questions associated with the enactment of any software. It allows the determination of the strengths and weaknesses of the current processes and allows us to evaluate the quality of specific processes and products (Van Solingen et al., 2002). A particular measurement/evaluation is useful only if it helps you to understand the underlying process or one of its resultant products. In turn, recognizing improvement of the process and products can occur only when the project has clearly defined goals for process and products. In other words, you cannot tell if you are going in the right direction until you determine your destination. (Fenton and Pfleeger, 1998).

According to (Fenton and Pfleeger, 1998) an evaluation strategy can be more successful if it is designed with the goals of the project in mind. One such strategy is the Goal Question Metric (GQM) approach, which is based upon the assumption that for an system to be measured in a focused way the goals of the system must be identified first, then those goals can be traced to the questions that are intended to answer those goals operationally. Finally provide a framework for interpreting the questions with respect to the stated goals in to measurable metrics. Thus it is important to make clear, at least in general terms, what the goals of each CA are so that these goals can be quantified whenever possible, and the quantified information can be analyzed as to whether or not the goals are achieved. The GQM approach proposed by Fenton and Pfleeger (1998) provides a framework involving three steps:

GOAL - List the major goals of the system.

QUESTION - Derive from each goal the questions that must be answered to determine if the goals are being met. Questions try to characterize the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected

viewpoint. Once the questions have been developed, the next step involves associating the question with appropriate metrics that will help in answering the question. **METRIC** - Decide what must be measured in order to be able to answer the questions adequately. A set of metrics is associated with every question in order to answer it in a quantitative way. The metrics can be classified as either:

Objective: If they depend only on the object that is being measured and not on the viewpoint from which they are taken; e.g., number of versions of a document, staff hours spent on a task, size of a program.

Subjective: If they depend on both the object that is being measured and the viewpoint from which they are taken; e.g., readability of a text, level of user satisfaction. (Fenton and Pfleeger, 1998; Van Solingen et al., 2002).

The GQM model is a top down hierarchical model as illustrated in Figure 1, the top level starts with a goal (specifying purpose of measurement, object to be measured, issue to be measured, and viewpoint from which the measure is taken).

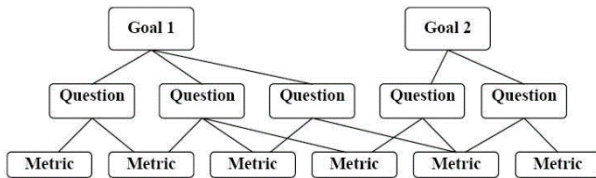


Figure 1: GQM Model

The goal is refined into several questions that usually break down the issue into its major components. Each question is a metric, some of them objective, some of them subjective. The same metric can be used in order to answer different questions under the same goal (Van Solingen et al., 2002).

This new CA evaluation framework was adopted for the evaluation of two novel conversational agent systems, UMAIR (Kaleem et al., 2014) and Abdullah CITS (Alobaidi et al., 2013).

3.1 Application of new Framework for the evaluation of the UMAIR CA

UMAIR was developed to serve as a customer service agent for the National Database and Registration Authority (NADRA) of Pakistan. The objective of UMAIR's development was to guide users through the complex process of ID card and Passport application. UMAIR conversed with the user in Urdu and was the first Urdu CA developed requiring a new CA architecture, using novel language processing techniques and algorithms. UMAIR was evaluated by utilizing proposed framework to formulate which metrics need to be evaluated to determine the success and robustness of the system and its newly developed algorithms. The results of applying the new evaluation framework to establish evaluation metrics for UMAIR are illustrated in Figure 2.

3.2 Application of new Framework for the evaluation of Abdullah CITS

The Abdullah CITS was developed to serve as an online tutor that teaches young children topics related to Islam. The aim of Abdullah was to mimic a human tutor by utilizing several teaching methodologies to deliver the tutorial through conversation with the users. Abdullah employed novel methods of detecting the users' level of knowledge and learning styles in order to adapt the tutorial

conversation to suit that individual users' ability and learning style (Alobaidi et al., 2013). One of the aims behind the evaluation of the Abdullah CITS was to verify if Abdullah was an effective tutor. The results of applying the new evaluation framework to establish evaluation metrics for gauging the effectiveness of Abdullah CITS as a tutor are illustrated in Figure 3.

GOAL - Implement an Effective Urdu CA

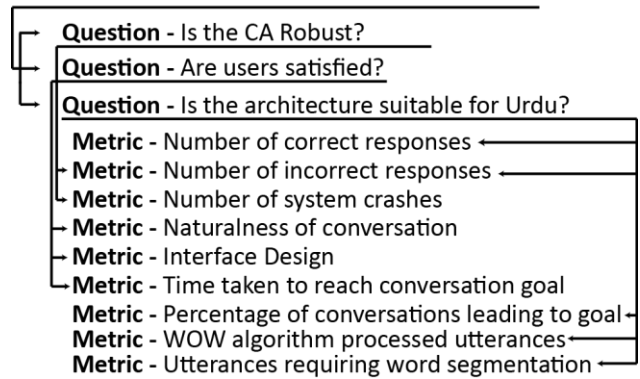


Figure 2 - UMAIR GQM Diagram

GOAL - To verify Abdullah CITS leads to satisfactory learning results

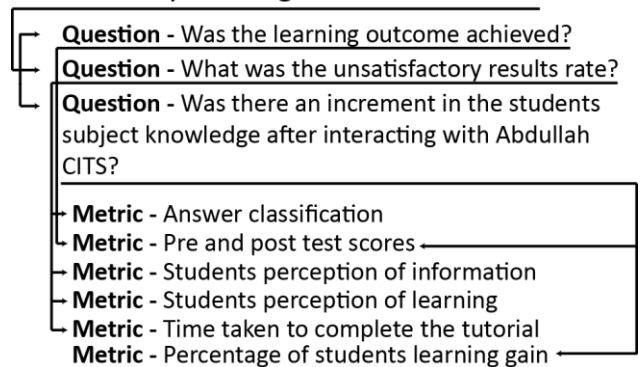


Figure 3 - Abdullah CITS GQM Diagram

4 Discussion

The results show that when the GQM method is applied to formulate the evaluation metrics for two different CAs the metrics derived suit the aims and objectives of that individual CA. It can be seen from the results that when GQM was applied to derive evaluation metrics for the UMAIR CA which is the first Urdu CA developed, the metrics derived are related to gauging the success of the components that make up the architecture of the CA. Whereas when GQM was applied to the Abdullah CITS the evaluation metrics derived were more related to the educational capabilities of the CA rather than measuring the success of the architecture components (i.e. was Abdullah an effective tutor). These metrics were utilized in the evaluation strategies of both the CAs by their respective researchers. Subsequent to the metrics being formulated they can be categorized in to their respective groups (i.e. subjective/objective) based on how the practitioner plans to capture the metrics for evaluation. This is traditionally done either through questionnaires for subjective metrics and some sort of log that captures the conversation and statistics related to the CA architecture and its components.

One of the advantages of the GQM approach is that multiple goals can be defined prior to evaluation and the

metrics can be formulated and categorized, which allows the pre-planning of how the metrics will be captured. One metric may be used to answer more than one question, therefore making the evaluation a more systematic process. Although there may be some overlap in common metrics such as conversation duration/time and conversation length etc. the metrics derived through GQM method are largely related to the development goals of the individual CA.

5 Conclusion

We have reported on the deployment of an adaptable framework for assessing CA quality in two distinctly different contexts. A new CA evaluation framework which is based on existing methods applied in a new context has been devised and tested which addresses the gap in current research related to the development and subsequent evaluation of natural language systems in general. The framework comprises of CA evaluation from an objective as well as subjective perspective in order to give an overall performance related CA evaluation. The proposed framework focuses on evaluating metrics related to the CAs ability to achieve the objective of its development by employing software evaluation methodologies (GQM). This approach allows CAs to be tested on an individual basis, meaning the metrics that are tested from system to system are derived based on the context of the systems implementation, thus allowing the evaluation metrics to be different depending on the development goals of the system being tested. Moreover it becomes easier to pre-determine better evaluation metrics when the proposed framework is used. The framework can be utilized by future research and practitioners to evaluate developed CAs, as the methodology is adaptable to suit individual CA development goals.

6 References

- Alobaidi, O. G., Crockett, K. A., O'shea, J. D. & Jarad, T. M. (2013) Abdullah: An Intelligent Arabic Conversational Tutoring System for Modern Islamic Education. Proceedings of the World Congress on Engineering.
- Artstein, R., Gandhe, S., Gerten, J., Leuski, A. & Traum, D. (2009) Semi-formal evaluation of conversational characters. *Languages: From Formal to Natural*. Springer.
- Fenton, N. E. & Pfleeger, S. L. (1998) *Software metrics: a rigorous and practical approach*, PWS Publishing Co.
- Hone, K. S. & Graham, R. (2000) Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, 6, 287-303.
- Ieee (2000) IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE Std 1471-2000*, i-23.
- Kaleem, M., O'shea, J. D. & Crockett, K. A. (2014) Word order variation and string similarity algorithm to reduce pattern scripting in pattern matching conversational agents. Computational Intelligence (UKCI), 2014 14th UK Workshop on. IEEE, 1-8.
- Keeling, K., Beatty, S., Mcgoldrick, P. & Macaulay, L. (2004) Face Value? Customer views of appropriate formats for embodied conversational agents (ECAs) in online retailing. System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on. IEEE, 10 pp.
- Latham, A., Crockett, K. & Mclean, D. (2014) An adaptation algorithm for an intelligent natural language tutoring system. *Computers & Education*, 71, 97-110.
- Latham, A., Crockett, K. A. & Bandar, Z. (2010) A Conversational Expert System Supporting Bullying and Harassment Policies. ICAART (1). 163-168.
- Lester, J., Branting, K. & Mott, B. (2004) Conversational agents. *The Practical Handbook of Internet Computing*.
- Martinez, F. F., Blázquez, J., Ferreiros, J., Barra, R., Macias-Guarasa, J. & Lucas-Cuesta, J. M. (2008) Evaluation of a spoken dialogue system for controlling a hifi audio system. Spoken Language Technology Workshop, 2008. SLT 2008. IEEE. IEEE, 137-140.
- Moller, S., Engelbrecht, K.-P., Kuhnel, C., Wechsung, I. & Weiss, B. (2009) A taxonomy of quality of service and quality of experience of multimodal human-machine interaction. Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on. IEEE, 7-12.
- O'shea, J., Bandar, Z. & Crockett, K. (2011) Systems Engineering and Conversational Agents. In: TOLK, A. & JAIN, L. (eds.) *Intelligence-Based Systems Engineering*. Springer Berlin Heidelberg.
- Rauschenberger, M., Schrepp, M., Cota, M. P., Olschner, S. & Thomaschewski, J. (2013) Efficient measurement of the user experience of interactive products. How to use the user experience questionnaire (ueq). example: spanish language version. *IJIMAI*, 2, 39-45.
- Roy, B. & Graham, T. N. (2008) Methods for evaluating software architecture: A survey. *School of Computing TR*, 545, 82.
- Shimazu, H. (2002) ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. *Artificial Intelligence Review*, 18, 223-244.
- Silvervarg, A. & Jönsson, A. (2011) Subjective and objective evaluation of conversational agents in learning environments for young teenagers. 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Barcelona, Spain.
- Skantze, G. & Hjalmarsson, A. (2013) Towards incremental speech generation in conversational systems. *Computer Speech & Language*, 27, 243-262.
- Turing, A. M. (1950) Computing machinery and intelligence. *Mind*, 433-460.
- Turunen, M., Hakulinen, J. & Kainulainen, A. (2006) Evaluation of a spoken dialogue system with usability tests and long-term pilot studies: similarities and differences. INTERSPEECH.
- Van Solingen, R., Basili, V., Caldiera, G. & Rombach, H. D. (2002) Goal Question Metric (GQM) Approach. *Encyclopedia of Software Engineering*.
- Walker, M. A., Litman, D. J., Kamm, C. A. & Abella, A. (1997) PARADISE: A framework for evaluating spoken dialogue agents. Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics. 271-280.

Tracking Conversational Gestures of Extraverts and Introverts in Multimodal Interaction

David Novick, Adriana Camacho, Ivan Gris, Laura M. Rodriguez

Department of Computer Science, The University of Texas at El Paso
500 W. University Ave., El Paso, TX, USA

novick@utep.edu, caro4874@gmail.com, ivangris4@gmail.com, lmrodriguez3@miners.utep.edu

Abstract

Much of our current research explores differences between extraverts and introverts in their perception and production of gestures in multimodal interaction with embodied conversational agent (ECA). While several excellent corpora of conversational gestures have been collected, these corpora do not distinguish conversants by personality dimensions. To enable study of the differences in distribution of conversational gestures between extraverts and introverts, we tracked and automatically annotated gestures of 59 subjects interacting with an ECA in the “Survival on Jungle Island” immersive multimodal adventure. Our work in developing provides an initial corpus for analysis of gesture differences, based on Jung’s four personality-type dimensions, of humans interacting with an ECA and suggests that it may be feasible to annotate gestures automatically in real time, based on a gesture lexicon. Preliminary analysis of the automated annotations suggests that introverts more frequently performed gestures in the gesture lexicon than did extraverts.

Keywords: dialog, grounding, feedback, gaze, nod

1. Introduction

Coordination of gesture is one of the principal factors in establishing rapport (Tickle-Degnen & Rosenthal, 1990). For embodied conversational agents (ECAs), coordination of gesture remains difficult; while humans can always be responsive to agents’ behaviors, ideally agents would also coordinate their own gestures autonomously to those of their human conversational partners.

In recent studies (e.g., Novick & Gris, 2014), we have been exploring the relationship between gesture and rapport in interaction between humans and ECAs, with particular attention to the differences in the paralinguistic behaviors of extraverts and introverts. These studies examined the differences between extraverts and introverts with respect to ECA-induced gestures (Gris, 2015) and interaction with ECAs who spoke with typically extraverted or introverted speech (Brixey, 2015). We leveraged part of these studies to record and classify the subjects’ conversational gestures. In this paper, we describe the purpose of the data collection, our technical approach to gesture-detection and annotation, and the corpus generated through two studies of human-ECA interaction in a multimodal immersive environment, and we discuss our current research using this corpus.

2. Gesture Corpora and Annotation

Several excellent corpora of conversational gestures have been collected. The Bielefeld Speech and Gesture Alignment Corpus (Lücking et al., 2013) comprised 25 dyadic route-and-landmark-description dialogues. The D64 Multimodal Conversational Corpus (Oertel et al., 2013) comprised two four-hour multiparty informal social conversations. The Headtalk (Carter & Adolphs, 2008) corpus comprised five hours of dyadic face-to-face conversational episodes. Researchers have also analyzed non-published corpora, such as videotapes of 28 dyads,

comprising more than five hours of dialogue, in a direction-giving task (Kopp, Tepper, & Cassell, 2004). These corpora, unfortunately, were not sufficient for the purposes of our personality-based analyses because the subjects had not been assessed on personality dimensions, such as extraversion. Also, for reliable statistical analysis, we were looking for an order of magnitude greater time of annotated recordings.

Annotation of gesture corpora has typically been conducted by hand-coding of the conversants’ behaviors. For example, a fine-grained approach was used in (Lücking et al., 2013), with gestural description that included both detailed body morphology and temporal phases, termed strokes. In a study of head nods as back-channel behaviors (Carter & Adolphs, 2008), the investigators again annotated the conversants’ behaviors manually. For an analysis of 18 minutes of a television conversation between two speakers, the researchers manually annotated, via Anvil, using a gesture lexicon (Kipp, Neff & Albrecht, 2007). This gesture-lexicon approach to annotation represented a middle ground between fine-grained but highly expensive coding and conversation-analytic but more subjective coding. The concept of a gesture lexicon raises the possibility of automated annotation of gesture: if target gestures can be represented so that they can be identified accurately by a gesture recognizer, then the process of coding gestures can be delegated to the recognizer. This approach is particularly useful when facing tens of hours of recordings of multimodal interaction.

3. Corpus Collection

Given our need for a corpus where the conversants could be classified by personality dimensions and our need to annotate many hours of data, for our study of differences between extraverts and introverts with respect to conversational gestures we leveraged the data collection for (Gris, 2015) to produce an automatically annotated

corpus of human gestures in multimodal interaction with an ECA. We also have non-annotated video recordings from (Brixey, 2015).

The interaction recorded in the corpus is of human-ECA multimodal interaction in “Survival on Jungle Island,” an immersive adventure intended for the study of human-ECA rapport (Kipp, Neff & Albrecht, 2007). Before each session, the subject completed an online instrument that profiled the subject’s personality in the four standard Jungian dimensions:¹

- Extraverted-introverted
- Sensing-intuition
- Thinking-feeling
- Judging-perceiving.

The subjects were students at the University of Texas at El Paso, primarily lower-division computer-science majors. Some subjects participated for credit in classes; others were recruited through posters such as the one in Figure 1.

The interaction took place in a room 17 feet long by 14 feet wide, where the agent’s environment was projected on an entire 14-foot-wide wall. The human, standing about eight feet in front of the projection, was recorded by a camera that was partially hidden in an artificial tree at the lower left corner of the projection wall. Other jungle-looking plants were arranged in front of the participant to add to the feeling of immersion. Figure 2 shows the immersion lab with the scenery and the projected agent. A Microsoft Kinect, which tracked the human’s gestures and recognized the human’s speech, was placed on the floor at the bottom center of the projection wall. We wrote custom applications (Novick et al., 2015) that manage the speech recognition and gesture detection and that classify, log, and plot the human’s gestures on a timeline of the interaction.

For the gesture recognition, the Kinect is able to recognize 20 major joints from the human body, and these are used to configure the subject’s position and posture, which in turn is defined as a pose. Because the positions of the joints are represented as Cartesian coordinates, their values will change depending on the position of the participant, even when performing the same pose. To accommodate this variation, each joint is triangulated using a base joint, switching from location of the Cartesian coordinates to the angles between the joints, and gestures were defined as sequences of poses (Gris, Camacho, & Novick 2015). To annotate the subject’s gestures, each pose was compared with the pose objects stored in our system’s library, which is an array of objects that contained relevant angles of joints representing a pose. For example, a crossed-arms pose used both the left and right arm joints, so this pose was defined with a margin of error that accounts for the minor differences of different people crossing their arms.

¹ The first author notes with relief that, based on this test, his own personality type is commonly described as making him one of the “benevolent ‘pedagogues’ of humanity.”

Fig 1. Poster for recruiting subjects.



Fig 2. Scenery and projected ECA. In this view, the ECA is an early version; in the actual sessions, the ECA was that shown in the images in Figure 1. The video camera was placed behind the small tree at the lower left of the screen, and the Kinect is visible as the small dark area below the center of the projection.

The poses in the library were defined with the UTEP gesture tool (Gris, Camacho, & Novick 2015). For the “Jungle” sessions, the library contained four “task-related” (i.e., iconic) gesture (defined by 7 poses) and “background” (i.e., beat and deictic) gestures (defined by 13 poses), as described in Table 1.

Figure 3 shows the gesture tool capturing an instance of the task-related high-five gesture. Figures 4a and 4b show anonymized examples of a task-related gesture (spear throw) and background gesture (crossed arms), respectively, from frames from the actual video

recordings of two subjects.

The study in (Gris, 2015) focused on iconic gestures, and these were tracked by the system to generate the ECA’s responses. The study did not use the background gestures for the interaction, but our system recorded, classified, and logged them for analysis of gestures with respect to personality differences. An analysis of the iconic gestures indicates that the automated recognition was accurate, on average, 59.4% of the time, with recognition rates for the seven separate gestures ranging from 25.0% to 77.8%.

Gesture	Description
Task-related	
Lift-hand*	Stretch arm in front with hand open; subject grabs agent’s hand
Striking	A hand above the other to rub sharply; subject lights fire by striking kindling
Spear throw*	Lift arm L-shaped; detected when fishing
Ventilate*	Lift forearm and move open hand side to side; used to reanimate fire with either right or left hand
Background	
Arms up	Detected when subject tries to signal helicopter
High-five*	Detected when agent celebrates with subject by simulating slapping subject’s hand high in the air
Normal stance	Arms on the side
Hands on hip	
Crossed arms	
Touch shoulder*	Hand to opposite shoulder
Hands together	Holding hands together in front
Hand on head#	Touching face or head
Balancing	Detected when walking on a tree-trunk bridge

Table 1. Gesture lexicon. * represents two poses, with either right or left arm/hand. # represents three poses, with either right, left, or both hands.



Fig 3. Using the UTEP gesture tool to capture an instance of the task-related “high five” gesture. The highlighted arms in the schematic human figure at left indicate that only the arms are being tracked for this definition, so that the gesture can be recognized independently of the position of the other parts of the subject’s body. The “Jungle” study had 80 subjects, and we obtained

usable recordings and annotations for 59 of these, ranging from 33.5 to 52 minutes each. The average session recording is just under 41 minutes. The total corpus of recordings is 37 hours and 29 minutes.



Fig 4a. Frame of corpus video, showing subject performing task-related “spear throw” gesture.



Fig 4b. Frame of corpus video, showing subject performing background “crossed arms” gesture.

The subjects’ personality profiles indicated that subjects included 32 extraverts, 24 introverts, and 3 persons with balanced extraversion-introversion scores. The minimum score was -67%, the maximum was 75%, the mean was 4.36%, and the standard deviation was 35.09%. Figure 5 presents a distribution histogram of the extraversion-introversion scores.

The automated annotations of gestures indicated that the subjects performed a total of 13,899 instances of gestures in the lexicon. Perhaps counter-intuitively, introverts performed, on average, 281.4 gestures per session, far more than the extraverts, who performed, on average, 223.3 gestures per session. The ranges of gesture frequency, though, were quite similar, as indicated by the summary data in Table 2.

4. Discussion

Our work in developing this corpus provides an initial body of recordings for analyzing gesture differences between extraverts and introverts when interacting with an ECA. And because our pre-session survey covered all four Jung personality-type dimensions, it would be possible to analyze the subjects’ gestures with respect to

differences in the sensing-intuition, thinking-feeling, and judging-perceiving dimensions as well. Our work also demonstrated an initial feasibility for automatic real-time annotation of gestures based on a gesture lexicon, which should enable analysis of much more extensive gesture corpora.

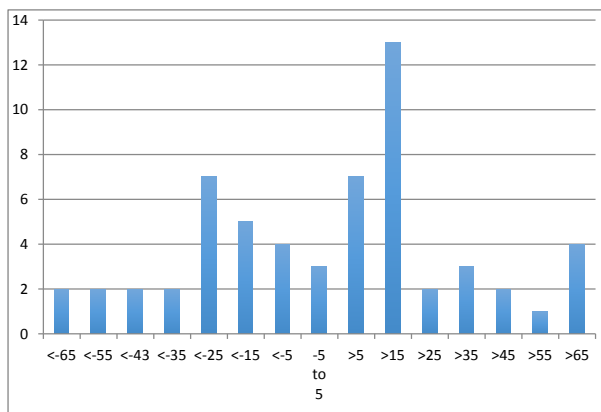


Fig 5. Distribution histogram of subjects' scores on the extraversion-introversion scale. Scores shown are differences of extraversion – introversion scores. Low scores indicate introversion, and high scores indicate extraversion.

	Extraverts	Introverts	Combined
N	32	24	56
Min	61	41	41
Max	873	840	873
Mean	223.3	281.4	248.2
Total	7146	6753	13899

Table 2. Summary data of gestures recognized.

However, our results are subject to several limitations, some of which are serious. One of those limitations is corpus synchronicity: while the automated tools and the agent sensors logged each event with a precision up to a millisecond, the additional high-definition video camera could not be started concurrently, leading to an offset between video, depth perception, and speech and gesture recognition logs. Fortunately, we were able to correct this manually.

Also, the design of the immersion space presented an illumination problem, especially for darker scenes where the subject's silhouette is not visible for short periods of time. Fortunately, the Kinect sensor performs its depth analysis through the use of infrared light, which meant that the gesture recognition was not affected even though validation via the video recordings was more difficult.

More serious, the lexicon of the poses/gestures detected by the system is limited. Our current library includes generally described gestures (e.g., a hand on the head), so if the subject performed another gesture (e.g., kneeling down) the system would not be able to recognize it. This could be addressed by adding additional poses/gestures to the lexicon, based on comparison of the

video recordings to the gesture logs. However, significantly increasing the size of the gesture lexicon might lead to lower recognition rates.

Another serious concern involves the potential for correlation between gestures in the lexicon and the system's recognition rate. We had observed for the task-related gestures that the recognition rate ranged from about 25% to over 75%. Because these differences in recognition appears to be systematic, analysis of gesture counts and frequencies could be systematically skewed. That is, gestures with lower recognition rates could be under-reported relative to gestures with higher recognition rates. The extent of this skew could be estimated by comparing the automated gesture logs with manual annotations from the video recordings. But because our analysis is based on the automated gesture annotations, which had high error rates for some of the gestures, drawing more solid conclusions will depend on significant improvement in the reliability of the gesture recognizer.

5. Conclusion

In developing this corpus, we sought to enable analysis of the paralinguistic behaviors of participants interacting with our agent, and to measure how these behaviors correlate with rapport-building between humans and agents for both extraverts and introverts. The corpus contains 37-and-a-half hours of full-body, standing, nonverbal behaviors, generally normal, unconscious gestures (denominated as "background gestures") and poses over a relatively long period of interaction. Our recognition system automatically annotated gestures from a lexicon of 20 gestures, but recognition accuracy, evaluated for the "task-related" gestures, ranged from 25 to 78 percent. Preliminary analysis of the automated annotations suggests that introverts more frequently performed gestures in the gesture lexicon than did extraverts.

A key challenge in building rapport through coordination of gestures is that, if done mechanically, the process can become apparent to the human, leading to a misconception that the agent is in fact an avatar and that the human can control the agent's movement directly. A related challenge involves the gestural production that amounts to direct mimicry, in which subtlety is lost and the human's awareness of the agent's limited gesture set can lead to a perceived state of system malfunction or even mockery. Accordingly, this corpus forms an initial step toward a pose and gesture library, and a gesture generation system using the library that appropriately coordinates gestures in real time, for both extraverts and introverts. Through understanding the frequency, type, and amplitude of gestures that humans unconsciously perform with agents we can improve the paralinguistic behavioral models of our agents, creating realistic, more natural behavioral sets that react appropriately to people.

6. Acknowledgments

We thank Alex Rayon, Victoria Bravo, Timothy

Gonzalez, Alfonso Peralta, Yahaira Reyes, and Diego A. Rivera for conducting many of the interaction sessions. We also thank the reviewers, whose comments and suggestions strengthened this paper.

7. References

- Brixey, J. (2015). Virtual rapport with extraverted agents. Masters' thesis, The University of Texas at El Paso.
- Carter, R., & Adolphs, S. (2008). Linking the verbal and visual: new directions for corpus linguistics. *Language and Computers*, 64(1), 275-291.
- Gris, I. (2015). Physical engagement as a way to increase emotional rapport in interactions with embodied conversational agents. Doctoral dissertation, The University of Texas at El Paso.
- Gris, I., Camacho, A., & Novick, D. (2015). Full-body gesture recognition for embodied conversational agents: The UTEP AGENT gesture tool, *Conference on Gesture and Speech in Interaction*, 131-136.
- Kipp, M., Neff, M., & Albrecht, I. (2007). An annotation scheme for conversational gestures: how to economically capture timing and form. *Language Resources and Evaluation*, 41(3-4), 325-339.
- Kopp, S., Tepper, P., & Cassell, J. (2004, October). Towards integrated microplanning of language and iconic gesture for multimodal output. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, 97-104.
- Lücking, A., Bergman, K., Hahn, F., Kopp, S., & Rieser, H. (2013). Data-based analysis of speech and gesture: The Bielefeld Speech and Gesture Alignment Corpus (SaGA) and its applications. *Journal on Multimodal User Interfaces*, 7(1-2), 5-18.
- Novick, D., Gris, I. (2014). Building rapport between human and ECA: A pilot study, *HCI International 2014*, Heraklion, Greece, July 2014.
- Novick, D., Gris, I., Rivera, D.A., Camacho, A., Rayon, A., & Gutierrez, M. (2015). The UTEP AGENT System. In *proc. 17th ACM Intl. Conf. on Multimodal Interaction*, 383-384.
- Oertel, C., Cummins, F., Edlund, J., Wagner, P., & Campbell, N. (2013). D64: A corpus of richly recorded conversational interaction. *Journal on Multimodal User Interfaces*, 7(1-2), 19-28.
- Tickle-Degnen, L. & Rosenthal, R. (1990). The Nature of rapport and its nonverbal correlates. *Psychological Inquiry*, 1:285-293.

Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora

Bayan AbuShawar, Eric Atwell

IT department; School of Computing
Arab Open University; University of Leeds
Amman, Jordan; leeds, Uk

b_shawar@aou.edu.jo; eric@comp.leeds.ac.uk;scmss@leeds.ac.uk

Abstract

A chatbot is a conversational agent that interacts with the users turn by turn using natural language. Different chatbots or human-computer dialogue systems have been developed using spoken or text communication and have been applied in different domains such as: linguistic research, language education, customer service, web site help, and for fun. However, most chatbots are restricted to knowledge that is manually “hand coded” in their files, and to a specific natural language which is written or spoken. This paper presents the program we developed to convert a machine readable text (corpus) to a specific chatbot format, which is then used to retrain a chatbot and generate a chat which is closer to human language. Different corpora were used: dialogue corpora such as the British National Corpus of English (BNC); the holy book of Islam Qur’an which is a monologue corpus where verse and following verse are turns; and the FAQ where questions and answers are pair of turns. The main goal of this automation process is the ability to generate different chatbot prototypes that spoke different languages based on corpus.

KeyWords:Chatbot, ALICE, AIML, Corpus

1. Introduction

Human machine conversation is a new technology integrating different areas where the core is the language, and the computational methodologies, which aim to facilitate communication between users and computers via natural language. A related term to machine conversation is the chatbot, which is a conversational agent that interacts with users turn by turn using natural language; chatbots have been applied in different domains such as: linguistic research, language education, customer service, website help, and for fun. The purpose of a chatbot system is to simulate a human conversation; the chatbot architecture integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language.

The idea of chatbot systems originated in the Massachusetts Institute of Technology (Weizenbaum 1966, 1967), where Weizenbaum implemented the Eliza chatbot to emulate a psychotherapist.

The idea was simple and based on keyword matching. The input is inspected for the presence of a keyword. If such a word is found, the sentence is mapped according to a rule associated with the keyword; if not, a connected free remark, or under certain conditions an earlier transformation, is retrieved. For example, if the input includes the keyword “mother”, ELIZA can respond “Tell me more about your family”. This rule is inspired by the

theory that mother and family are central to psychological problems, so a therapist should encourage the patient to open up about their family; but the ELIZA program does not really ‘understand’ this psychological strategy, it merely matches the keyword and regurgitates a standard response. To keep the conversation going, ELIZA has to produce responses which encourage the patient to reflect and introspect, and this is done mechanically using some fixed phrases if no keyword match is found such as “Very interesting. Please go on.”, or “Can you think of a special example?” The next major program was PARRY (Colby 1973, 1999). In contrast to ELIZA, instead of simulating a psychotherapist, PARRY modelled a paranoid patient during an interview with his therapist. Saygin, Cicekli, and Akman (2000) noticed that, “Both ELIZA and PARRY use certain tricks to be able to successfully perform in conversations. ELIZA directs the conversation away from herself by asking questions. ELIZA uses parts of the user’s input in the output questions and seems to be following the conversations. In addition to these techniques, PARRY has little stories to tell and tend to insert these in the conversation.” Figure 1 shows a sample of chatting with ELIZA.

The seventies and eighties, before the arrival of graphical user interfaces, saw rapid growth in text and natural-language interface research. Since that time, a range of new chatbot architectures have been developed, such as: MegaHAL (Hutchens, 1996), CONVERSE (Batacharia et al., 1999), ELIZABETH (Abu Shawar and Atwell, 2002),

HEXBOT (2004) and ALICE (2002). With the improvement of data-mining and machine-learning techniques, better decision-making capabilities, availability of corpora, robust linguistic annotations/processing tools standards like XML and its applications, chatbots have become more practical, with many commercial applications. In this paper, ALICE/AIML chatbot architecture and the pattern matching techniques used within it are presented in section 2. Section 3 illustrates our development of a Java program that can convert a machine readable text (corpus) to the AIML format used by ALICE. A description of the automatic extraction of training data from different corpora is explained in section 4. The results and conclusion are presented in sections 5 and 6 consequently.

Human: Men are all alike.

ELIZA: In what way?

Human: They're always bugging us about something.

ELIZA: Can you think of a special example?

Human: Well, my boyfriend made me come here.

ELIZA: Your boyfriend made you come here?

Human: He says I'm depressed much of the time.

ELIZA: I'm sorry to hear that you are depressed much of the time.

Human: It's true. I'm unhappy.

ELIZA: Do you think coming here will help you not to be unhappy?

Human: I need some help, that much seems certain.

Figure 1. A sample of conversation with ELIZA (from Jurafsky and Martin 2000)

2. ALICE Architecture

We have worked with the ALICE open-source chatbot initiative. ALICE is the Artificial Linguistic Internet Computer Entity (ALICE 2002, Wallace 2003), originated by Wallace in 1995. A sample of chatting with ALICE is shown in figure 2. ALICE stores knowledge about English conversation patterns in AIML files. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML). It was developed by the Alicebot free software community during 1995-2000 to enable people to input dialogue pattern knowledge into chatbots based on the ALICE free software technology.

User: Hello, my name is George and I am the human.

ALICE: Hi George.

User: Would you like to play a game?

ALICE: Sure.

User: Ok then. I will email it to you by use of the CD drive.

ALICE: What hobbies do you enjoy?

User: I like cinema.

ALICE: What types of films?

User: All.

ALICE: What movie did you see this year that you liked?

User: I don't have time for movies.

ALICE: I thought you liked cinema.

User: I did, but then you started to say stupid things.

Figure 2. A sample of conversation with ALICE (from Loebner prize contest 2004)

AIML consists of data objects called AIML objects, which are made up of units called topics and categories as shown in figure 3. The topic is an optional top-level element, it has a name attribute and a set of categories related to that topic. Categories are the basic unit of knowledge in AIML. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which represents the user input, and a template, which implies the ALICE robot answer. The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant. The idea of the pattern matching technique is based on finding the best, longest, pattern match.

```
<aiml version="1.0">
<topic name="the topic">
<category>
<pattern>USER INPUT</pattern>
<template>Chatbotanswer</template>
</category></topic></aiml>
```

Figure 3. The AIML format

2.1 Types of ALICE/AIML Categories

There are three types of the AIML categories: atomic categories, default categories, and recursive categories.

Atomic categories are those with patterns that do not have wildcard symbols, `_` and `*`, e.g.:

```
<category>
<pattern>WHAT IS 2 AND 2</pattern>
<template> It is 4 </template>
</category>
```

In the above category, if the user inputs ‘What is 2 and 2’, then ALICE answers ‘it is 4’.

Default categories are those with patterns having wildcard symbols `*` or `_`. The wildcard symbols match any input but they differ in their alphabetical order. Assuming the previous input WHAT IS 2 AND 2, if the robot does not find the previous category with an atomic pattern, then it will try to find a category with a default pattern such as:

```
<category>
<pattern>WHAT IS 2 *</pattern>
<template><random>
<li>Two.</li>
<li>Four.</li>

<li>Six.</li>
</random></template>

</category>
```

So ALICE will pick a random answer from the list. The `_` works on the same manner but if first words are missed: `<template>_4</template>` so any template end with 4 will match

Recursive categories are those with templates having `<srai>` and `<sr>` tags, which refer to simply recursive artificial intelligence, and symbolic reduction. Recursive categories have many applications: symbolic reduction that reduces complex grammatical forms to simpler ones; divide and conquer that splits an input into two or more subparts, and combines the responses to each; and dealing with synonyms by mapping different ways of saying the same thing to the same reply as the following example:

```
<category>
<pattern>HALO</pattern>
<template><srai>Hello</srai>
</template></category>
```

The input is mapped to another form, which has the same meaning.

2.2 ALICE/AIML Pattern Matching Technique

The AIML interpreter tries to match word by word to obtain the longest pattern match, as this is normally the best one. This behavior can be described in terms of the Graphmaster as shown in figure 4. A Graphmaster is a set of files and directories, which has a set of nodes called nodemappers and branches representing the first words of all patterns and wildcard symbols. Assume the user input starts with word X and the root of this tree structure is a folder of the file system that contains all patterns and templates; the pattern matching algorithm uses depth first search techniques:

- If the folder has a subfolder starting with underscore then turn to, “`_/`”, scan through it to match all words suffixed X, if no match then:
- Go back to folder, try to find a subfolder starts with word X, if so turn to “`X/`”, scan for matching the tail of X, if no match then:
- Go back to the folder, try to find a subfolder start with star notation, if so, turn to “`*/`”, try all remaining suffixes of input following “X” to see if one match. If no match was found, change directory back to the parent of this folder, and put “X” back on the head of the input. When a match is found, the process stops, and the template that belongs to that category is processed by the interpreter to construct the output.

The above paragraph describes how internally ALICE search a response for the user input, how to match between user input and the stored knowledge in AIML brain. Users does not know what knowledge is there, but whatever the user input is, ALICE will try to find the longest pattern match based on lexical matching. In the following section we will clarify how we implement a Java program to read from any corpus and convert it into AIML format then extend ALICE knowledge by the generated categories.

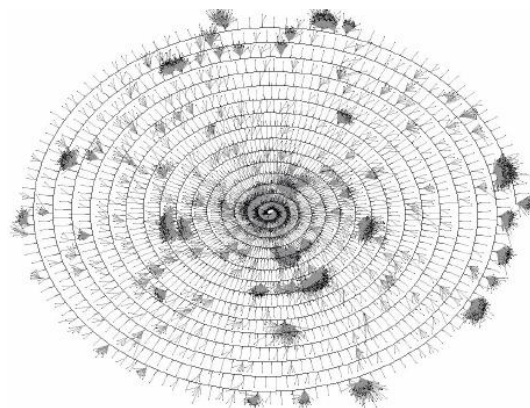


Figure 4. A Graphmaster that represents ALICE brain

3. Automatic Generation of AIML Categories

We developed a java program that converts the readable text (corpus) to the chatbot language model format. The aim of this software is create ALICE knowledge base automatically and based on specific corpus or domain. Then extend current knowledge of ALICE with the new generated files. Two versions of the program were generated. The first version is based on simple pattern template category, so the first turn of the speech is the pattern to be matched with the user input, and the second is the template that holds the robot answer. Usually the dialogue corpora contain linguistic annotation that appears during the spoken conversation such as overlapping, and using some linguistic filler. To handle the linguistic annotations and fillers, the program is composed of four phases as follows:

1. Phase One: Read the dialogue text from the corpus and insert it in a vector.
2. Phase Two: Text reprocessing modules, where all linguistic annotations such as overlapping, fillers and other linguistic annotations are filtered.
3. Phase Three: converter module, where the pre-processed text is passed to the converter to consider the first turn as a pattern and the second as a template. Removing all punctuation from the patterns and converting it to upper case is done during this phase.
4. Phase Four: Copy these atomic categories in an AIML file.
5. Phase Five: Building a frequency list of patterns lexical. This list will be used to obtain the first and second most significant words (least frequent words) from each utterance.
6. Phase Six: Building the default category file. AIML pattern-matching rules, known as "categories", are created. There are two possible types of match: input matches a complete pattern so atomic categories will be matched; or input matches 1st or 2nd most significant word in the user input (least frequent words).

After building the atomic files in phase 4, the program is adopted to a more general approach to finding the best match against user input from the learned dialogue. In case no exact matching is found the default categories are built to give a close answer based on significant keywords: first word and most significant ones.

A restructuring module was added to map all patterns with the same response to one form, and to transfer all repeated pattern with different templates to one pattern with a random list of different responses. Two machine learning approaches were adapted to build default categories (phase six) as follows:

- First word approach, based on the generalisation that the first word of an utterance may be a good clue to an appropriate response: if we cannot match the whole input utterance, then at least we can try matching just the first word. For each atomic pattern, we generated a

default version that holds the first word followed by wildcard to match any text, and then associated it with the same atomic template.

- Most significant word approach, we look for the word in the utterance with the highest "information content", the word that is most specific to this utterance compared to other utterances in the corpus. This should be the word that has the lowest frequency in the rest of the corpus. We choose the most significant approach to generate the default categories, because usually in human dialogues the intent of the speakers is hiding in the least-frequent, highest-information word. We extracted a local least frequent list from the corpus, and then compared it with each token in the pattern to specify the first most significant word within that pattern. Later on, the second most significant word were also used in conjunction with first word and first most significant word to obtain the best pattern match. Once may argue that significant word could misspelled, in this case a default category will be built but it will be matched if user input have the same misspelled word which will be rare.

The program was enhanced to handle different format and structure for three main types of corpora as follows:

- *Dialogue corpora*: where each corpus has its own annotations, so filtering process will differ. And the first utterance is considered as a pattern and the next one as a template (response).
- *Monologue corpora*: represented by the holy book of Islam the Qur'an where each verse is considered as a pattern and the next one as a template.
- *FAQ corpora*: where the question represents the pattern and the answer represents the template.

4. Applying the Program on Multi Corpora

During the enhancement and evolving to our system, we tried different types of corpora: human dialogue transcripts, monologue and structural one (FAQs, QA). In this section a brief discussion of all corpora used and how our software was evolved are presented.

4.1 Human Dialogue Transcripts

Two versions of the system were initially developed. The first version is based on simple pattern template category, so the first turn of the speech is the pattern to be matched with the user input, and the second is the template that holds the robot answer. This version was tested using the English-language Dialogue Diversity Corpus (DDC, Mann, 2002). This corpus is a collection of links to different dialogue corpora in different fields, where each corpus has its own annotation format. After text re-processing and filtering, the Java program was simple and considered each utterance as a pattern, and its successor as a template that represents chatbot answer. This experiment reveals the

problems of utilising dialogue corpora such as: long turns; no standard annotations to distinguish between speakers, overlapping and irregular turn taking, and using linguistic fillers. (Abu Shawar and Atwell 2003a). Unfortunately most of these problems also occur in other corpora, which necessitate changing the filtering process to meet the difference in the corpora format. Figure 5 shows a sample of DDC and its equivalent atomic category.

```
<u who=F72PS002>
<s n="32"><w ITJ>Hello<c PUN>.
</u>
<u who=PS000>
<s n="33"><w ITJ>Hello <w NP0>Donald<c PUN>.
</u>
```

The corresponding AIML atomic category is:

```
<category>
<pattern>HELLO</pattern>
<template>Hello Donald</template>
</category>
```

Figure 5. A sample of DDC turn and its equivalent atomic category

To prove that our system can be used with other dialogue corpora, the Minnesota French Dialogue Corpus (Kerr 1983) was used. One advantage of the Machine-Learning approach to re-training ALICE is that we can automatically build AIML from a corpus even if we don't understand the domain or even the language; to demonstrate this, the program was tested using the Corpus of Spoken Afrikaans (van Rooy, 2003). The new chatbot that speaks Afrikaans was published on line using Pandorabot service¹, and we encouraged open-ended testing and feedback from remote users in South Africa; this allowed us to refine the system more effectively. We adopted three evaluation metrics (AbuShawar and Atwell, 2003b, 2007).

- *Dialogue efficiency in terms of matching type*: whether a user input is matched an atomic category, or first word, and most significant ones. For this purpose four sample of dialogues were analyzed. Outputs illustrates that the first word and the most significant approach increase the ability to generate answers to users and let the conversation continue.
- *Dialogue quality metrics based on response type*: where the responses of the four dialogue sample are classified according to an independent human evaluation of "reasonableness": reasonable reply, weird

but understandable, or nonsensical reply. We gave the transcript to an Afrikaans-speaking teacher and asked her to mark each response according to these classes. The number of turns in each dialogue and the frequencies of each response type were estimated. For this evaluator, it seems that "nonsensical" responses are more likely than reasonable or understandable but weird answers.

- *Users' satisfaction assessment based on an open-ended request for feedback*: expanding the AIML pattern matching using the first-word and least-frequent-word approaches yielded more favorable feedback. Our informants found the conversations less repetitive and more interesting.

It took several years for the Alice Botmaster to accumulate the 50,000 categories in the current public-domain set of AIML files (Wallace 2003). The British National Corpus (BNC 2002) was selected to test if our system is capable to generate more than one million categories extracted from the BNC. The British National Corpus (BNC) (Aston and Burnard 1998) is a collection of text samples amounting to over 100 million words, extracted from 4,124 modern British English texts of all kinds, both spoken and written. The corpus is annotated using SGML (XML-like) mark-up, including CLAWS Part-of-Speech category of every word. All annotations are marked between <angle brackets>. The corpus is partitioned into two types: the spoken and the written transcripts. In order to generate versions of ALICE which talk like humans, we decided to retrain ALICE using the BNC spoken transcripts comprising 10 million words. A range of different chatbot systems were generated using the program that converts the BNC to AIML format; this enabled us to animate and explore the language of various BNC sample texts. 1,153,129 categories were created in total, and divided in files representing a range of domains such as: sport, food, travel and media. By the BNC prototypes the chatbot could be used a tool to visualise a large corpus. Figure 6 shows a sample of chatting with BNC prototype in the sport domain.

Human: Hi, do you like sport?

Chatbot: Hi there. No I'm not much into sports. I like solo sports like rollerblading and skiing though.

Human: Do you know anything about football?

Chatbot: Any of several games played with an inflated leather ball by two teams on a field with goals at each end, the object being to get the ball across the opponents' goal.

Human: What about women's football?

Chatbot: I think football seems to, to bring out the worst in a lot of people and that's the part of it that I do not like

¹<http://www.pandorabots.com/pandora>

Human: Do you like it?

Chatbot: and er, I think until women's football take it on board all the way erm, they're not really gonna stand up with the men they have to become, referees, they have to become coaches, and more and more of these people erm, taking responsibility I go along and watch women's football and get really infuriated to see a man out there refereeing the game, and often in a very condescending manner! Instead of carding someone he'll put his arm round her and say Do not !It's not very nice to do that !Do not do that again!

Human: It is an enjoyable game.

Chatbot:Aha Aha It is a case is not it, that if you want to be a er a professional woman footballer, certainly there's more opportunities on the continent than there are in Britain, is, is is that the case?Yes?

Human: I watch it regularly.

Chatbot:Because if football's on it has to be watched!It's like a a religion! Fi

Figure 6. A sample of chat with BNC trained Sports chatbot

4.2 Holy Qur'an Text

In order to see if we can extract AIML files from a corpus text which is not a dialogue where no turns are there, another version of the system was updated to generate an Arabic AIML files extracted from the Qur'an, the holy book of Islam. Moslems believe the Arabic text is a faithful transcription of the infallible words of God relayed through the angel Gabriel to the prophet Mohammed, who memorised the entire monologue to pass on verbally. Mohammed's successors transcribed the message to simplify transmission and avoid corruption, but every Moslem should aim to memorise it, in original Arabic, and to use the Qur'an to guide every aspect of their lives. The Qur'an consists of 114 sooras, which could be considered as sections, grouped into 30 parts (chapters). Each soora consists of more than one verse (Ayya). These ayyas are sorted, and must be shown in the same sequence. The AIML-learning system was revised to handle the non-conversational nature of the Qur'an. Aversion of English/Arabic was built where in this case if an input was a whole verse (Ayya), the response will be the next verse of the same soora; or if an input was a question or a statement, the output will be all verses which seem appropriate based on the significant word. As ample of chatting with the Qur'an chatbot is presented in figure 7.

THE DAYBREAK, DAWN, CHAPTER NO. 113

With the Name of Allah, the Merciful Benefactor, the Merciful Redeemer

113/1 Say: I seek refuge with the Lord of the Dawn

113/2 From the mischief of created things;

113/3 From the mischief of Darkness as it overspreads;

113/4 From the mischief of those who practice secret arts;

113/5 And from the mischief of the envious one as he practices envy.

سورة الفلق (113)

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

قُلْ اَعُوْذُ بِاللّٰهِ مِنَ الْغَلَقِ {113/1} مِنْ شَرِّ مَا خَلَقَ {113/2} وَمِنْ شَرِّ غَاسِقٍ اِذَا وَقَبَ {113/3} وَمِنْ شَرِّ النَّفّٰثَاتِ الْيَهُودِ الْعُقَدِ {113/4} وَمِنْ شَرِّ حَاسِدٍ اِذَا حَسَدَ {113/5}

.... <http://www.islam4u.com> مركز الاشعاع الاسلامي ...
(صفحة 604)

Figure 7. Samples of the Arabic and English versions of the Qur'an

To measure the quality of the answers of the Qur'an chatbot version, the following approach was applied:

1. Random sentences from Islamic sites were selected and used as inputs of the English/Arabic version of the Qur'an.
2. The resulting transcripts which have 67 turns were given to 5 Muslims and 6 non-Muslims students, who were asked to label each turn in terms of:
 - o Related (R), in case the answer was correct and in the same topic as the input.
 - o Partially related (PR), in case the answer was not correct, but in the same topic.
 - o Not related (NR), in case the answer was not correct and in a different topic.

Proportions of each label and each class of users (Muslims and non-Muslims) were calculated as the total number over number of users times number of turns. Four out of the 67 turns returned no answers, therefore actually 63 turns.

In the transcripts used, more than half of the results were not related to their inputs. A small difference can be noticed between Muslims and non-Muslims proportions. Approximately one half of answers in the sample were not related from non-Muslims' point of view, whereas this figure is 58% from the Muslims' perspective. Explanation for this includes:

- The different interpretation of the answers. The Qur'an uses traditional Arabic language, which is sometimes difficult to understand without knowing the meaning of some words, and the historical story behind each verse.
- The English translation of the Qur'an is not enough to judge if the verse is related or not, especially given that non-Muslims do not have the background knowledge of the Qur'an.

Using chatting to access the Qur'an looks like the use of a standard Qur'an search tool. In fact it is totally different; a searching tool usually matches words not statements. For example, if the input is: "How shall I pray?" using chatting: the robot will give you all ayyas where the word "pray" is found because it is the most significant word. However, using a search tool² will not give you any match. If the input was just the word "pray", using chatting will give you the same answer as the previous, and the searching tool will provide all ayyas that have "pray" as a string or substring, so words such as: "praying, prayed, etc." will match.

Another important difference is that in the search tool there is a link between any word and the document it is in, but in the chatting system there is a link just for the most significant words, so if it happened that the input statement involves a significant word(s), a match will be found, otherwise the chatbot answer will be: "I have no answer for that". (Abu Shawar and Atwell 2004a, 2004b).

4.3 FAQ Websites

Corpus Linguists have shown increasing interest in possibilities of using the Web as corpus source, so we decided to try using FAQs as pseudo-dialogue to train chatbots. In these versions the chatbot expects user input similar to FAQ questions, and replies with FAQ answers, or with lists of links to possible FAQ answers. FAQs have clear turn-taking structure, and are widely used in computing, so are a good source of training corpus data. The system was adapted such that the question represents a pattern, and the answer represents the template in building atomic AIML files. The frequency list was constructed from questions (patterns).

A Frequently Asked Questions document or FAQ is a structured database; most of the "data-cleaning" problems found with dialogue corpora (such as overlapping, and more than two speakers) are not found in the FAQs. Moreover almost all HTML tags embedded in the FAQ file are recognised by the ALICE interpreter, because the AIML definition allows HTML tags to be embedded within templates. The questions and answers were extracted directly from the HTML files of the FAQ.

Different categories are added to extend the chance of finding answers, where the answer is either a set of links in case most significant words are found in more than one question or a direct answer in the instance where only one match was found. In addition to first word and most significant word (1st), we extracted second most significant one (2nd) (least frequent words). For each significant word, four default categories were added to handle different position of word in pattern; another category holding first

word, 1st or 2nd most significant as appeared in original question was generated.

At the end a FAQchat prototype was generated and tested against Google by nine staff members and twelve students at School of Computing in University of Leeds. As a result, 68% overall of our sample of users (21 users) managed to find answers using the FAQchat; while 46% found it by Google. Since there is no specific format to ask the question, there are cases where some users could find answers while others could not. In terms of preferences, 51% of the staff, 41% of the students, and 47% overall preferred using FAQchat against 11% who preferred the Google (Abu Shawar et al., 2005).

Two example English FAQ chatbots were generated which are trained using: the FAQ of the School of Computing (SoC) at the University of Leeds; and the public-domain Python programming language FAQ website. The Python FAQ website has been translated into other languages, so we were able to generate a Spanish-speaking Python advisor. Figure 8 illustrates the questions and answers generated with these FAQ chatbots.

a. Using the FAQ of the School of Computing at the University of Leeds

Human: How to remove a job from the printer?

Ask me: If you want to remove a job from the printer queue then find out the job number using the [lpq command](#), and in a UNIX shell type: `%lprm -Pprinternamejob_no`

You can only remove jobs that you have sent to the printer yourself - apart from Support that is.

For more information look at:

[How do I remove a job from the printer queue? \(Printing \)](#)

b. Using the FAQ of the Python programming language website

Human: what is a class?

English_Python_FAQ: A class is the particular object type created by executing a class statement. Class objects are used as templates to create instance objects, which embody both the data (attributes) and code (methods) specific to a datatype. A class can be based on one or more other classes, called its base class(es). It then inherits the attributes and methods of its base classes.....

Figure 8. A sample of questions with answers generated by the FAQ chatbot versions

²<http://www.islamicity.com/QuranSearch/>

The success with using chatbot as a tool to answer SoC FAQs encouraged us to try other FAQs, or Questions Answers (QA) corpora to investigate the possibility of using a chatbot as a tool to access an information portal without the need for sophisticated natural language processing or logical inference. In 2008, an open ended FAQChat was built where the knowledge base was extracted from multiple FAQs: Perl, Linux and Python. In 2010, TREC09 QA track was used to retrain ALICE and in 2011 Arabic QA corpora was used. Overall User trials with AskJeeves, Google and generated chatbot demonstrate that chatbot is a viable alternative, and in fact many users prefer it to Google as tool to access FAQ databases (Abu Shawar 2008, 2011), (AbuShawar and Atwell 2010). We managed to demonstrate that simple ALICE-style chatbot engine could be used as a tool to access the WWW FAQs, or QAs. No need for sophisticated natural language analysis or logical inference; a simple (but large) set of pattern-template matching rules will be sufficient.

5. Results and Outputs

The automatic system was evolved within each corpora used, the system was able to generate AIML training data for ALICEchatbot to serve in different domains. For example, the number of training data that were extracted automatically from the BNC corpus is 1,153,129 which is 20 times bigger than any existing chatbot: the large public domain. Some of the different usages of ALICE, and some statistics about the number of categories that were built automatically are presented in tables 1 and 2 consequently.

ALICE chatbot	Purpose
ALICE (Wallace 2003)	Entertainment chatbot
Speak2Me(2004)	A web-based version of ALICE aimed at Chinese learners of English, allowing them to practise chatting to a well-spoken young lady, a virtual British English native speaker
AfrikaanaChatbot	A tool to learn/practice a language.
BNCChatbot	A tool to visualize (animate) a corpora.
Arabic/English Qura'nchatbot	A tool to learn Qur'an for Arabic and English speakers.
FAQchat	A tool to access an information portal

Table 1. A sample of services that can be provided by ALICE chatbot

<i>The number of AIML training data obtained from BNC domains</i>					
BNC domains	Atomic		Default		Total
	Categories	File size (KB)	Categories	File size (KB)	Categories
Sport	3283	974	7913	1,820	11196
World affairs	3120	983	8756	1,886	11876
Travel	640	314	1636	575	2276
Media	1061	491	3126	1,210	4187
Food	503	93	1125	168	1628
<i>The number of AIML training data obtained from the Qur'an</i>					
The Holy book of Islam the Qur'an	Atomic		Default		Total
	Categories	File size (MB)	Categories	File size (MB)	Categories
	22,950	11,51	74,349	44.2	79,299
<i>The number of AIML training data obtained from the Arabic FAQ Websites</i>					
Arabic FAQ domains	Atomic Categories	Default Categories	Total Categories		
Mother&kids	164	2,057	2,221		
Blood_disease	97	1,292	1,389		
Blood_charity	69	960	1,029		
Health&fasting	25	318	343		
Teeth_care_issu	42	614	656		

Table 2. A sample of AIML training data obtained from different corpora

6. Conclusion

A chatbot is a conversational agent that interacts with users using natural language. An illustration of ALICE chatbot in terms of: Knowledge base and its pattern matching technique is presented. The main lack in ALICE and other chatbots is the manual developing of its knowledge, we managed to build a software program that reads from a corpus and converts it to ALICE knowledge base. This paper presents the algorithm of the automation process and machine learning techniques used to extract training data from a corpus. Different corpora were used to retrain ALICE that reveals other usefulness for of chatbot rather than an entertainment tools. A chatbot could be used as a tool to animate or visualise a corpus, a tool to learn/practice English, Arabic, Afrikaans or other languages, and a tool to access an information portal to provide answers to questions.

7. References

- Abu Shawar, B., and Atwell, E. (2002). A comparison between Alice and Elizabeth chatbot systems. School of Computing research report 2002.19. Leeds: University of Leeds.
- Abu Shawar B. and Atwell E. (2003a). Using dialogue corpora to retrain a chatbot system. In Archer, D., Rayson, P., Wilson, A. and McEnery, T. (eds.), *Proceedings of the Corpus Linguistics 2003 conference (CL2003)*. Lancaster University, UK, pp. 681-690.
- Abu Shawar B. and Atwell E. (2003b). Using the Corpus of Spoken Afrikaans to generate an Afrikaans chatbot. *SALALS Journal: Southern African Linguistics and Applied Language Studies*. Vol. 21, pp. 283-294.
- Abu Shawar B., and Atwell E. (2004a). An Arabic chatbot giving answers from the Qur'an / Un chatbot arabe qui donne des reponses du Coran. In B. Bel & I. Marlien (Eds.), *Proceedings of TALN2004: XI Conference sur le Traitement Automatique des Langues Naturelles*, (Volume 2, pp. 197-202). ATALA.
- Abu Shawar B., and Atwell E. (2004b). Accessing an Information system by chatting. In F. Mezziane & E. Metais (Eds.) *Natural Language Processing and Information Systems: Proceedings of NLDB04* (pp. 407-412). Berlin: Springer-Verlag
- Abu Shawar B., Atwell E., and Roberts A. (2005). FAQChat as an Information Retrieval System. In: Vetulani, Zygmunt (ed.) *Human Language Technologies as a Challenge*. Proceedings of the 2nd Language and Technology Conference, WydawnictwoPoznanskie, Poznan, Poland, pp.274-278.
- Abu Shawar B., and Atwell E. (2007). *Different measurement metrics to evaluate a chatbot system*. in Proceedings of the **NAACL'07** Workshop: Bridging the Gap: Academic and Industrial Research in Dialog Technologies. Pp.89-96, ACL.
- Abu Shawar, B. (2008). Chatbots are natural web interface to information portals. In proceedings of INFOS2008, Pp. NLP101-NLP107
- Abu Shawar, B., and Atwell, E. (2010). Chatbots: Can they serve as natural language interfaces to QA corpus? In Proceeding of the sixth IASTED International Conference Advances in Computer Science and Engineering (ACSE 2010). Pp. 183-188
- Abu Shawar, B. (2011). A Chatbot as a natural web Interface to Arabic web QA. *International Journal of Emerging Technologies in Education (iJET)*, Vol. 6, No. 1, pp.37-43.
- Alice (2002). A.L.I.C.E AI Foundation, [Online:] <http://www.Alicebot.org/>
- Aston, G., and Lou, B. (1998). *The BNC Handbook: Exploring the British National Corpus with SARA*. Edinburgh: Edinburgh University Press.
- Batacharia, B., Levy, D., Catizone, R., Krotov, A. & Wilks, Y. (1999). CONVERSE: a conversational companion. In Y. Wilks (Ed.) *Machine conversations* (pp. 205-215). Boston/Dordrecht/London: Kluwer.
- BNC (2002). British National Corpus website <http://www.natcorp.ox.ac.uk/>
- Chai J. and Lin J. (2001). "The role of a natural language conversational interface in online sales: a case study". *International Journal Of Speech Technology*, vol 4, pp. 285-295.
- Chai J., Horvath V., Nicolov N., Stys-Budzikowska M., Kambhatla N., Zadrozny W. (2000). "Natural language sales assistant – A web-based dialog system for online sales". Proceedings of thirteenth annual conference on innovative applications of artificial intelligence, 2000.
- Chatbot. 2015. [online]: <https://www.chatbots.org/>
- Colby, K. (1973). Simulation of belief systems. In R. Schank & K. Colby (Eds.) *Computer models of thought and language* (pp. 251-286). San Francisco: Freeman.

- Colby, K. (1999). Human-computer conversation in a cognitive therapy program. In Wilks, Y. (eds.) *Machine conversations*. Kluwer, Boston/Dordrecht/London. Pp. 9-19.
- HEXBOT (2004). HEXBOT chatbot website. <http://www.hexbot.com/>
- Hutchens, J. (1996). How to pass the Turing test by cheating. School of Electrical, Electronic and Computer Engineering research report TR97-05. Perth: University of Western Australia.
- Kerr, B. (1983). *Minnesota Corpus*. Minneapolis: University of Minnesota Graduate School.
- Mann, W. (2002). *Dialog Diversity Corpus*. [Online]: <http://www/rfc.usc.edu/~billmann/diversity/DDivers-site.htm>
- Saygin, A., Cicekli, I. & Akman, V. (2000). Turing test: 50 years later. *Minds and Machines*, 10 (4), 463-518.
- Van Rooy, B. (2003). *Transkripsiehandleiding van die Korpus Gesproke Afrikaans (Transcription Manual of the Corpus of Spoken Afrikaans)*. Potchefstroom: Potchefstroom University.
- Voth D. (2005). Practical agents help out. *IEEE intelligent systems*, pp. 4-7. [Online]: <http://csdl2.computer.org/comp/mags/ex/2005/02/x2004.pdf>
- Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*. Vol. 10, No. 8, pp. 36-45.
- Weizenbaum J. (1967). Contextual understanding by computers. *Communications of the ACM*. Vol. 10, No. 8, pp. 474-480.
- Wallace, R. (2003) *The elements of AIML style*. ALICE AI Foundation.

Shared Task on Data Collection and Annotation

RE-WOCHAT 2016 – SHARED TASK DESCRIPTION REPORT

Luis F. D'Haro

Human Language Technology,
Institute for Infocomm Research
luisdhe@i2r.a-star.edu.sg

Bayan Abu Shawar

IT department; School of Computing,
Arab Open University
bshawar@yahoo.com

Zhou Yu

Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, 15217
zhouyu@cs.cmu.edu

Abstract

This report presents and describes the shared task on “Data Collection and Annotation” conducted with RE-WOCHAT, the first Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents - Development and Evaluation. We describe the main road map envisaged for this and future shared tasks, as well as the proposed collection and annotation schemes. We also summarize the result of the shared task in terms of chatbot platforms made available for it and the amount of collected chatting sessions and annotations.

Keywords: shared task, chat-oriented dialogue, data collection, manual annotation

1. Introduction

As part of the activities of the workshop, RE-WOCHAT¹ (Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents - Development and Evaluation) has accommodated a shared task on “Data Collection and Annotation”. The main objective of this shared task is to develop and test a new evaluation framework for non-goal-oriented dialogue engines.

The rest of the paper is structured as follows. First, a brief background to the shared task is presented in section 2, followed by the basic objectives and intended roadmap for the shared task in section 3. Then, in section 4 the chatbot platforms made available for the shared tasks are briefly introduced and finally, in section 5, a summary of the collected data and annotations are presented.

2. Shared Task Background

Different from task-oriented dialogue, automatic evaluation of chat-oriented dialogue poses some interesting challenges to due to the specific nature and lack of specific goals in it. Different approaches have been proposed to this end, including time of engagement and user satisfaction (Abu Shawar and Atwell, 2007), Dialogue Coherence Models (Gandhe and Traum, 2008), and comparative evaluations (Banchs and Kim, 2014).

Although data driven approaches have provided a useful means for training evaluation metrics in many other areas of research, one of the main problems related to the development of similar strategies for chat-oriented dialogue is for certain the lack of enough annotated data. In this sense, the main motivation for a shared task on “Data Collection and Annotation” is to provide an experimental platform for the research community to generate data and resources for chat-oriented dialogue

research. This must be achieved by a collaborative effort continued over time and expanded to multiple languages and modalities.

3. Main Objectives and Road Map

This shared task is part of a larger scope initiative, which main objectives are (1) collecting chat-oriented dialogue data that can be made available for research purposes and (2) developing a framework for the automatic evaluation of chat-oriented dialogue.

This effort comprises three interdependent tasks:

- **Task 1. Chat data collection:** participating teams will produce dialogues between human users and chat engines, as well as between humans only.
- **Task 2. Subjective evaluation:** participating teams will manually evaluate a selection of the generated dialogues according to different subjective evaluation metrics.
- **Task 3. Subrogated metrics:** participant teams will attempt to model the manually generated subjective evaluation metrics by using machine learning techniques

The current edition of the Shared Task in RE-WOCHAT has focused only on Tasks 1 and 2 described above. Task 3 will be addressed in future editions of the workshop after enough annotated data has been generated to make feasible the use of machine learning approaches.

Four different ways of participation in the shared tasks were defined:

- **Chatbot provider.** Participants owning a chatbot engine and willing to provide access to it either by distributing a standalone version of it or via a web service or web interface.
- **Data generator.** Participants willing to use one or more of the provided chatbots to generate dialogue sessions with it.

¹ <http://workshop.colips.org/re-wochat/shared.html>

- **Data provider.** Participants owning or having access to a chatbot that are not accessible to the general public willing to generate chatting sessions and share the generated data with other participants.
- **Data annotator.** Participants willing to annotate some of the generated and/or shared dialogue sessions by following the provided annotation guidelines.

A total of 14 volunteers registered for participating in the first edition of the shared task. These 14 volunteers accounted for a total of six chatbot providers, seven data generators, three data providers and eight data annotators.

4. Chatbot Platforms made Available

The six chatbot engines made available for the shared task include:

- **Joker.** An example-based system that uses a database of indexed dialogue examples automatically built from a television drama subtitle corpus to manage social open-domain dialogue (Dubuisson et al, 2016b).
- **IRIS.** Informal Response Interactive System, which implements a chat-oriented dialogue system based on the vector space model framework (Banchs and Li, 2016).
- **Py-Eliza.** A Python-based stand-alone version of the famous Eliza chatbot created by Weizenbaum in 1966 (D’Haro, 2016).
- **Sarah.** An upgraded version of Alice bot, developed by Dr.Wallace in 1995 (AbuShawar, 2016)
- **TickTock.** A chatbot with a goal to engage users in an everyday conversation. It is a keyword based retrieval system with engagement conversational strategies (Yu et al, 2016).
- **Politician.** A question-answering system, which is designed as a chatbot imitating a politician. It answers questions on political issues. (Kuboň and Hladká, 2016).

For more detailed information about each one of these chatbots, the reader can refer to the Shared Task Chatbot Description Reports in the Workshop Proceedings.

Most of these chatbots are available via online interfaces or as standalone systems for collecting chatting interactions with registered participants. The plan is to keep these systems available on a continuous basis and grow the number of systems on future editions of the shared task.

The following tips have been provided to the shared task participants to be taken into account during the data generation phase:

- Use the same nickname when interacting with the different chatbots. As chatting sessions are anonymous, this will be the only way to track all different chatting sessions for the same user.
- Remember these are just chatbots, do not expect

too much from them. Please try to converse as much as you can and in the most natural way.

- Generate as much chatting sessions as you can. Ideally, a chatting session should include more than 20 turns but no more than 50 turns.

5. Data Collection and Annotations

To the date this report was written, a total of 554 chatting sessions had been collected since the beginning of the Shared Task. In addition, a total of 61 contributed dialogs were received, out of which 41 dialogs were contributed by the Joker system² (Dubuisson et al, 2016a). All these provided chatbot systems are still available for the participants to continue interacting with them, so the data collection is still ongoing. An updated report will be presented the day of the RE-WOCHAT workshop and will be made available at the workshop official website.³

Table 1 and Figure 1 show some of the statistics of the dialogs with all the chatbots. Surprisingly chatbots use more vocabulary and longer sentences than humans. This may be because they want to keep engaged humans or because humans tend to be more specific to keep the dialog focused and easier for the chatbot to understand.

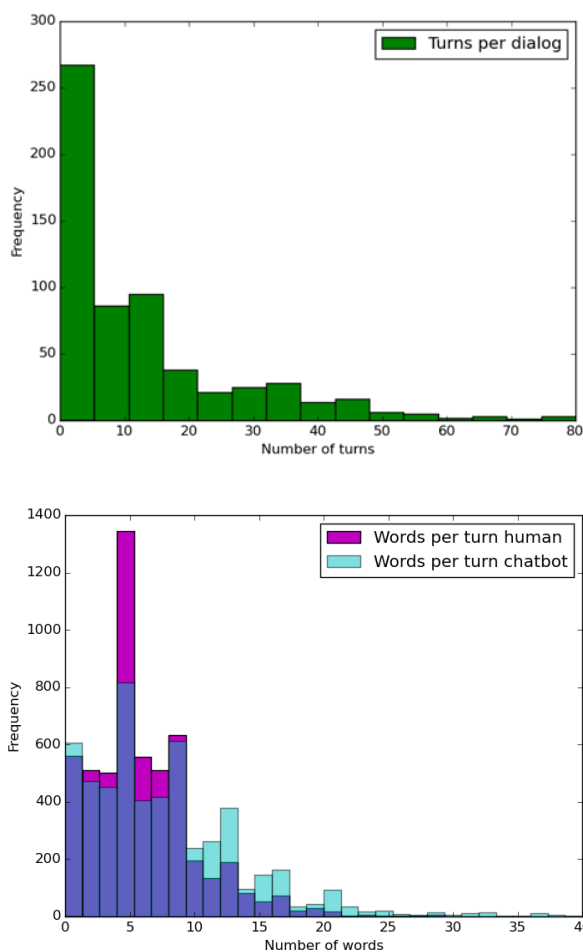


Figure 1. a) Histograms for number of turns per dialog and b) words per turn comparing human vs chatbot turns

² <https://ucar.limsi.fr>

³ <http://workshop.colips.org/re-wochat/index.html>

	Chatbot	Human	Total
No. dialogs	-	-	615
No. Turns	-	-	8589
Vocab. Size	4445	4088	
Av. no. Words per sentences	7.66 ±2.64	5.84 ± 0.40	
Polarity	0.08±0.25	0.07±0.28	
Subjectivity	0.22±0.31	0.23±0.33	

Table 1. Basic statistics of the collected dialogue sessions.

On the other hand, we also provide polarity and subjectivity calculated using TextBlob⁴, where polarity is within the range [-1.0, 1.0] and subjectivity is a within the range [0.0, 1.0], being 0.0 a very objective sentence and 1.0 a very subjective sentence. Here, we cannot see a dominant trend on either the chatbots or humans, but with a very small difference toward chatbots being more subjective and positive.

A total of 126 of the collected chatting sessions have been manually annotated by human evaluators according to the proposed subjective evaluation guidelines (see Table 2). These guide- lines comprise the assignment of one (out of three) basic subjective scores to each turn in a chatting session. The possible valid tags are: VALID, ACCEPTABLE and INVALID, meaning:

- **VALID:** this score is used to access a response that is semantically and pragmatically valid given the previous utterance as well as the previous recent dialogue context. Some examples of VALID responses to the utterance “how old are you?” include: “I am 25”, “older than you” and “I am quite young”.
- **ACCEPTABLE:** this score is used to access a response that is not necessarily semantically valid but can be acceptable, given the previous recent dialogue context, from the pragmatic point of view. Some examples of ACCEPTABLE responses to the utterance “how old are you?” include: “let us better talk about food”, “how old are you?” and “what did you say before?”
- **INVALID:** this score is used to access a response that is definitively invalid given the previous utterance and the recent dialogue context. Some examples of INVALID responses to the utterance “how old are you?” include: “he goes to the supermarket every Saturday” or “I like pizza”.

In additions to the three subjective scores described above, annotators were also requested to evaluate the polarity and offensiveness of the utterances, in those cases in which this was possible. These optional annotations were used according to the following conventions:

- **POSITIVE:** this tag might be used to indicate positive polarity of the response.
- **NEGATIVE:** this tag might be used to indicate negative polarity of the response.
- **OFFENSIVE:** this tag might be used to indicate

inappropriate offensive response, which does not necessarily contain swear words.

- **SWEARLANG:** this tag might be used to indicate the explicit presence of inappropriate language, regardless whether it is offensive or not.

In a similar way to the data collection task, annotations are scheduled to continue over time. An updated report will be presented the day of the RE-WOCHAT workshop and will be made available at the workshop website.

Metric	Chatbot	Human	Total	
No. of evaluated dialogs	126			
No. of evaluated turns	2723			
Subjective scores	Valid	777	1432	2209
	Acceptable	534	315	849
	Invalid	600	65	665
	<i>Kappa</i>	0.567		
Optional annotations	Positive	34	31	65
	Negative	59	57	116
	Offensive	50	32	82
	Swearlang	31	2	33

Table 2. Basic statistics of the annotated dialogue sessions.

6. Conclusions and Future Work

This report has presented and described the shared task on “Data Collection and Annotation” conducted with RE-WOCHAT, the first Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents - Development and Evaluation. We described the main road map envisaged for this and future shared tasks, as well as the proposed collection and annotation schemes used in the shared task activities. We also summarized the result of the shared task in terms of chatbot platforms made available for it and the amount of collected chatting sessions and annotations.

As future work for preparing next editions of the shared task, we plan to consolidate a data collection and annotation platform for chat-oriented dialogue, by centralizing the different available chatbots into the same interface. Additionally we plan to evaluate different gamification strategies to encourage more people to participate and contribute to the shared task activities.

7. Acknowledgements

We want to thank the workshop organizers: Rafael E. Banchs, Ryuichiro Higashinaka, Wolfgang Minker, Joseph Mariani and David Traum for their assistance and support during the organization of the shared task.

Similarly, we also want to thank all the volunteers who contributed to the shared task activities: Andreea Niculescu, Emer Gilmartin, Guillaume Dubuisson Duplessis, Kheng Hui Yeo, Natalia Kocyba, Rafael Banchs, Ryuichiro Higashinaka, Soe Gon Yee Thant, Sophie Rosset, Vincent Letard, and Vlad Maraev.

⁴ <https://textblob.readthedocs.org/en/dev/>

8. References

- AbuShawar, B., Atwell, E. (2007). Different measurement metrics to evaluate a chatbot system, in Proceedings of Workshop on “Bridging the Gap: Academic and Industrial Research in Dialogue Technologies”. pp. 89—96, NAACL-HLT
- AbuShawar, B. (2016) Sarah Chatbot, in Proceedings of RE-WOCHAT, LREC 2016, Shared Task Report.
- Banchs, R., Kim, S. (2014). An empirical evaluation of an IR-based strategy for chat-oriented dialogue systems, in Proceedings of APSIPA, Special Session on Chatbots and Conversational Agents
- Banchs, R., Li, H. (2016) IRIS – Informal Response Interactive System, in Proceedings of RE-WOCHAT, LREC 2016, Shared Task Report.
- D’Haro, L.F. (2016) Py-Eliza: A Python-based Implementation of the Famous Computer Therapist, in Proceedings of RE-WOCHAT, LREC 2016, Shared Task Report.
- Dubuisson Duplessis, G. and Letard, V. and Ligozat, A.-L. and Rosset, S. (2016a) Purely Corpus-based Automatic Conversation Authoring, in Proceedings 10th International Conference on Language Resources and Evaluation (LREC), May 2016. 8 p.
- Dubuisson Duplessis, G., Letard, V., Ligozat, A.L., Rosset, S. (2016b) Joker Chatterbot, in Proceedings of RE- WOCHAT, LREC 2016, Shared Task Report.
- Gandhe, S., Traum, D. (2008). An evaluation understudy for dialogue coherence models, in Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, pp. 172—181, ACL.
- Kuboň, D., Hladká, B. (2016). Politician, in Proceedings of RE-WOCHAT, LREC 2016, Shared Task Report.
- Yu, Z., Xu, Z., Black A.W., Rudnicky A.I. (2016) Tick Tock, in Proceedings of RE-WOCHAT, LREC 2016, Shared Task Report.

POLITICIAN

RE-WOCHAT 2016 - SHARED TASK

CHATBOT DESCRIPTION REPORT

David Kuboň

Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics,
Charles University in Prague
Malostranské náměstí 25
kubondavid@seznam.cz

Barbora Hladká

Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics,
Charles University in Prague
Malostranské náměstí 25
hladka@mff.cuni.cz

Abstract

We present a question-answering system *Politician* designed as a chatbot imitating a politician. It answers questions on political issues. The questions are analyzed using natural language processing techniques and no complex knowledge base is involved. The language used for the interviews is Czech.

1 Politician's General Description

The system *Politician* enables users to interview a virtual politician. It opens a conversation with a standard greeting *Hello!* followed up by the question *What are you interested in?* Then it is the user's turn and he asks a question. Once morphological and syntactic annotations are available for his question, an appropriate answer template is selected from a manually created set. Then *Politician* replies to the question. The conversation is closed when the user does not pose any other question.

2 Politician Technical Description

The *Politician* performs a series of three basic steps visualized in Figure 1:

1. **Analyzing user question** – we use a pipeline of natural language processing tools, namely a morphological analyzer, a tagger, a parser, and a named-entity recognizer. For Czech, the procedures implemented in the natural

language processing (NLP) framework *Treex* are applied, see (Popel and Žabokrtský, 2010).

2. **Selecting answer template** – we manually designed a list of answer templates. Thus our task is to select an answer template that fits the question the best.
3. **Generating *Politician's* reply** – we transform the answer from the step 2 into a grammatically correct reply.

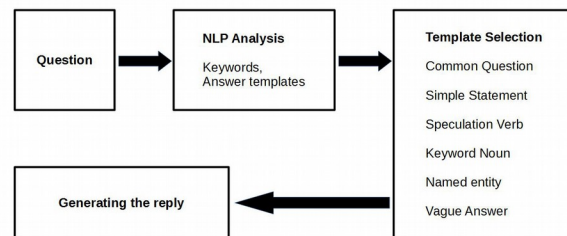


Figure 1: Chatbot overall architecture

Once the question is processed with NLP tools, the process of reply generation starts:

- Some basic criteria are applied to filter out common questions and simple statements.
- The nouns and verbs in the question are checked against the manually created list of keywords typical for present-day political conversations.
- The question is checked whether it consists of verbs of speculation such as *to think, to suppose, to believe*.
- If one of the detected keywords is the subject of the sentence or the subject attribute,

then it is chosen for an appropriate answer template selection. If neither is dependent on the subject, the choice of the keyword is made randomly.

- The named entities recognized in the question are analyzed.
- If all the preceding steps are considered failed, then a vague reply is provided. It is selected randomly from the set of replies prepared for such a situation. For example, *I analyzed these issues in detail in my press release yesterday and I do not intend to get back to them any longer.*
- When the answer template is selected, a grammatically correct reply is generated so that *Politician* inserts the keywords detected in the question into a reply in a proper grammatical form. For example, from the template *[keyword] [to be] [fruit] of our long-term efforts.* we can generate the reply *Reforms are the fruit of our long-term efforts.*

3 Sample Dialogue Sessions

We provide a sample conversation selected from the data collected at public promotion events. We use italics to emphasize the words, which were selected as the keywords for the answer template selection.

Table 1: Sample Session with the Politician

1	POLITICIAN: Hello! What are you interested in?
2	USER: Good afternoon. How much is the Czech state <i>debt</i> per capita?
3	POLITICIAN: This is a very good question. We have already voted to create a special committee that will deal with the debt. Let us wait for its conclusion.
4	USER: What impact will <i>inflation</i> have on Czech market?
5	POLITICIAN: This topic will be emphasized in the next election campaign.
6	USER: What is your opinion on amnesty granted by

	President <i>Václav Klaus</i> ?
7	POLITICIAN: I do not like this person at all.
8	USER: Do not you think that MPs' <i>salaries</i> are too high?
9	POLITICIAN: Unfortunately, judging this topic is not within my competence. I am sorry, but I have to go now, I have an important meeting.

4 Turing Test

So far we have been presenting *Politician* mainly for students e. g. attending a university open house. We were conducting the Turing test (Turing. 1950) during these events as follows:

- The human answerer was instructed to pose only vague and non-specific replies. The users were told to ask only political questions. If they did not follow these instructions, we did not include such conversations into a final evaluation.
- Each conversation contained four user questions.
- Each user did two interviews, one with the human and one with *Politician*. Then he judged in which of the two interviews he was interviewing the computer.

In total, we conducted over 30 conversations on political topics. The *Politician* was able to fool users more than 33% of the time. We do not provide exact number because the boundary between relevant and irrelevant conversations is highly disputable. One third of the users fooled were confused by misspelling errors, half of them considered *Politician's* replies more natural and the rest of them did not have any relevant reason.

References

- Popel, M., and Žabokrtský, Z. (2010). TectoMT: Modular NLP Framework. *In Proceedings of IceTAL, 7th International Conference on Natural Language Processing*, pages 293-304, Reykjavík, Iceland.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, pages 433-460.

JOKER CHATTERBOT

RE-WOCHAT 2016 - SHARED TASK

CHATBOT DESCRIPTION REPORT

Guillaume Dubuisson Duplessis LIMSI, CNRS, Université Paris-Saclay, F-91405 Orsay gdubuisson@limsi.fr	Vincent Letard LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, F- 91405 Orsay letard@limsi.fr	Anne-Laure Ligozat LIMSI, CNRS, ENSIIE, Université Paris-Saclay, F-91405 Orsay annlor@limsi.fr	Sophie Rosset LIMSI, CNRS, Université Paris- Saclay, F-91405 Orsay rosset@limsi.fr
--	--	---	--

Abstract

The Joker chatterbot is an example-based system that uses a database of indexed dialogue examples automatically built from a television drama subtitle corpus to manage social open-domain dialogue.

1 Joker Chatterbot General Description

The Joker chatterbot is part of the Joker project which aims at building a generic intelligent user interface providing a multimodal dialogue system with social communication skills including humor and other social behaviors (Devillers et al. 2015). This project is primarily interested in entertaining interactions occurring in a social environment (e.g., a cafeteria).

The Joker chatterbot targets dyadic social open-domain conversations between a human and the system. It is based on a conversational strategy that has been automatically authored from a corpus of television drama subtitles (Dubuisson Duplessis et al., 2016). It aims at generating surprising and entertaining contributions in response to a human utterance in order to maintain the human participation in the dialogue.

Dialogues with the chatterbot consist in social open-domain conversations carried out in English through a text-based web interface. Dialogues are structured as a sequence of pairs of utterances: one from the human followed by the response of the system.

2 Joker Chatterbot Technical Description

The conversational strategy behind the chatterbot belongs to the category of example-based dialogue modeling that aims at using a database of semantically indexed dialogue examples to manage dialogue (Lee et al., 2009). One feature of our approach is the complete automation of the conversation strategy authoring process from the creation of the database of dialogue examples based on a corpus of dialogues to the conversational management process, see (Dubuisson Duplessis et al., 2016) for a complete account.

The main purpose of the dialogue management process is to select an appropriate response from a database of dialogue examples given the human utterance, and adapt it by taking into account the current dialogue. To this end, our approach discerns three main steps: (1) the selection of candidate system responses from the database of examples, (2) the selection of the most appropriate response, and (3) the transformation of the selected response by taking into account the human utterance and the history of dialogue.

Our dialogue example database is built from subtitles coming from a variety of television drama genres including comedy and sci-fi series (e.g., “The Big Bang Theory”, “Malcolm In the Middle”, “Real Humans”). 274,227 dialogue examples have been extracted from a corpus of 432 subtitle files.

3 Sample Dialogue Sessions

Table 1 presents a sample dialogue session of real user interaction with the Joker chatterbot. It has been collected in the context of the RE-WOCHAT shared task via a web interface. This session shows the emergence of a dialogue that can be qualified as roughly coherent, despite the fact that our system focuses on the local coherence of dialogue. Indeed, it only takes into account the last user utterance to select its response.

This example shows a certain ability of the system to select surprising utterances (4, 6, 12). It also points out the selection of incoherent utterances such as 8 and 14. Additional examples and discussions about the capabilities of the system can be found in (Dubuisson Duplessis et al., 2016).

Table 1: Sample Session with the Joker Chatterbot

1	USER: hi there !
2	SYSTEM: hi
3	USER: how are you joker ?
4	SYSTEM: i 've got to get back to work
5	USER: yeah i 've got to get back to work also
6	SYSTEM: i 'll go with you
7	USER: yeah sure
8	SYSTEM: hey
9	USER: how are you feeling today ?
10	SYSTEM: good why
11	USER: i just want to know
12	SYSTEM: if you guys are organizing something
13	USER: no no , we are not organizing anything
14	SYSTEM: printing a story that has become a topic of conversation

4 Additional Information

This chatterbot system has been previously used to collect a corpus of 41 open-domain textual dialogues with 27 human participants (Dubuisson Duplessis et al., 2016). This corpus is freely available at the URL: <https://ucar.limsi.fr>. An evaluation of the system was carried out involving objective measures and subjective self-reports (Dubuisson Duplessis et al., 2016). This has made it possible to discuss the general capabilities of the system (regarding understandability, repetition and coherence of system responses) as well as the subjective evaluations of human participants (in terms of amusement, engagement and enjoyability).

The current version of the system is available online at the URL: <https://webjoker.limsi.fr/RE-WOCHAT>.

Acknowledgments

This work was partly funded by the JOKER project and supported by ERA-Net CHIST-ERA, and the “Agence Nationale pour la Recherche” (ANR, France).

References

- Devillers, L., Rosset, S., Dubuisson Duplessis, G., Sehili, M., Béchade, L., Delaborde, A., Gossart, C., Letard, V., Yang, F., Yemez, Y., Türker, B., Sezgin, M., El Haddad, K., Dupont, S., Luzzati, D., Estève, Y., Gilmartin, E., and Campbell, N. (2015). Multimodal data collection of human-robot humorous interactions in the joker project. In 6th International Conference on Affective Computing and Intelligent Interaction (ACII).
- Dubuisson Duplessis, G., Letard, V., Ligozat, A.-L., Rosset, S. (2016). Purely Corpus-based Automatic Conversation Authoring. In 10th International Conference on Language Resources and Evaluation (LREC).
- Lee, C., Jung, S., Kim, S., and Lee, G. G. (2009). Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51(5):466–484.

IRIS (INFORMAL RESPONSE INTERACTIVE SYSTEM) RE-WOCHAT 2016 - SHARED TASK CHATBOT DESCRIPTION REPORT

Rafael E. Banchs

Institute for Infocomm Research,
1 Fusionopolis Way, Singapore
rembanchs@i2r.a-star.edu.sg

Haizhou Li

Institute for Infocomm Research,
1 Fusionopolis Way, Singapore
hli@i2r.a-star.edu.sg

Abstract

This report describes IRIS (Informal Response Interactive System), a chat-oriented dialogue system based on the vector space model framework. IRIS was one of the systems made available as part of the RE-WOCHAT Shared Task platform for collecting human-chatbot dialogue sessions.

1 IRIS General Description

IRIS is a chatbot who is conversant in a large variety of topics. She has been “watching” movies for a while and has learned chatting patterns from the dialogues in the movies. She has such a very good memory that she is able to chat based on semantic associations she builds between a given chatting session and all the movies she “watched” before! Sometimes, she is also able to learn some basic concepts from the people she chats with.



Figure 1: IRIS Chatbot Avatar

IRIS is like a nice teenager that likes to chat with people, although sometimes she can lose her temper and become a little bit naughty. However, most of the time, she behaves like a good chatbot! IRIS impersonation is depicted in Figure 1.

2 IRIS Technical Description

IRIS is a data-driven chatbot engine (Banchs and Li, 2012) that has been trained with dialogues from movies (Banchs, 2012). IRIS response selection is based on a two-level semantic matching strategy (turn and dialogue history), which has been shown to outperform a baseline turn-level search strategy (Banchs and Kim, 2014). Semantic matching in IRIS is based on cosine similarity over the vector space model.

Figure 2 depicts a diagram of the overall system architecture for the IRIS chatbot system.

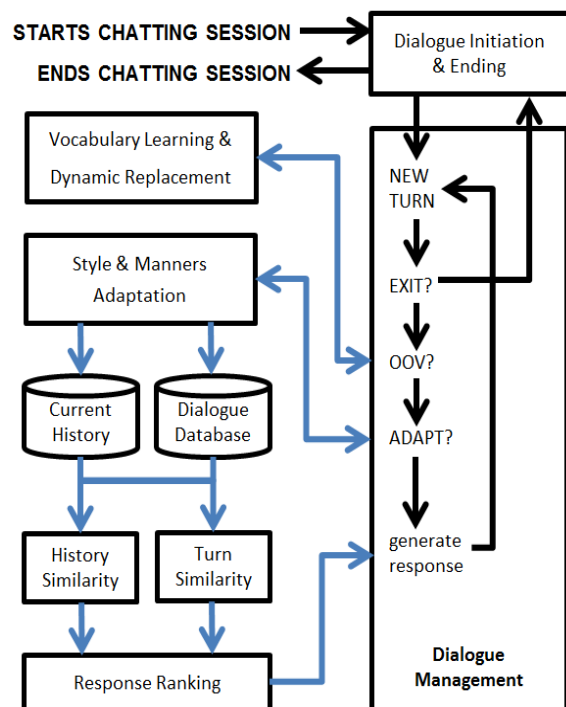


Figure 2: IRIS Overall System Architecture

As seen from the figure, in addition to its main dialogue management loop, IRIS has several different components. These include: (1) a rule-based initiation/ending module that allows for smoothly starting and ending chatting sessions; (2) a vocabulary learning module that allows for memorizing and reusing definitions that are provided by the users; (3) a style & manners adaptation component inspired on the Rocchio’s relevance feedback algorithm (Banchs and Li, 2012); and (4) the two-level semantic similarity matching and ranking module, which selects final system responses from the database of movie dialogues.

It is important to mention that IRIS implements a response raking strategy that favors recall. This allows for the system to increase the variability of its responses, as well as to change topics in a more natural way.

3 Sample Dialogue Session

For the purpose of RE-WOCHAT Shared Task on data collection and annotation, an online version of IRIS was made available. Table 1 presents one example of the collected chat sessions.

Table 1: Sample Session with IRIS

1	USER: hi iris, how old are you
2	IRIS: Thirty-three next month.
3	USER: wow that is great
4	IRIS: I know. Dreadful, isn't it?
5	USER: not sure about it
6	IRIS: Sure.
7	USER: ok, tell me a joke
8	IRIS: Ok.
9	USER: go ahead please
10	IRIS: Thank y... The line goes DEAD.
11	USER: ha ha ha
12	IRIS: No, have fun first. Then laugh. Now...

The implemented online version of IRIS is based on a JavaScript/PHP client that communicates with a standalone version of IRIS running as a server. The client’s URL is available through the workshop’s website: http://workshop.colips.org/re-wochat/chatbot_info.html. The current implementation is restricted to one single user at a time.

4 Data Collection in the Shared Task

Since the online version of IRIS has been available (until the time this report was written), a total of 104 chatting sessions have been collected. In total, these sessions comprise about 2.5K turns and 12.3K running words, corresponding to an average of 5 words per turn and 24 turns per session.

From all these sessions, 64 of them have been manually evaluated at the turn level by participants of the Shared Task. All collected sessions and the corresponding annotations will be made available to the research community in the workshop’s website. Both, data collection and data annotation, efforts will continue over time and will be reported in future editions of the workshop.

5 Future Work and Development

The main limitation of the current online version of IRIS is that it can only chat with one person at a time. In this sense, developing and multi-thread version of IRIS is one of our priorities.

Another important component to be improved is the response ranking module. A detailed analysis of IRIS’ lists of candidate responses has shown that the system can be significantly improved by better selecting the responses.

References

- Banchs, R.E. and Li, H. (2012) IRIS: a chat-oriented dialogue system based on the vector space model. In Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (ACL), demo systems
- Banchs, R.E. (2012) Movie-DiC: a movie dialogue corpus for research and development. In Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (ACL), short papers
- Banchs, R.E. and Kim, S. (2014) An empirical evaluation of an IR-based strategy for chat-oriented dialogue systems. In Proc. of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)

PyEliza: A Python-based implementation of the famous computer therapist

RE-WOCHAT 2016 - SHARED TASK CHATBOT DESCRIPTION REPORT

Luis Fernando D'Haro

Institute for Infocomm Research
One Fusionopolis Way #21-01 Connexis
Singapore, 138632

`luisdhe@i2r.a-star.edu.sg`

Abstract

In this report we provide information about the functionalities and capabilities of pyEliza a python-based implementation of the famous Eliza chatbot proposed by Weizenbaum in 1966. PyEliza implements a rule-based chatbot that encourage users to talk about their lives and feelings. In addition to the chat capabilities, this chatbot features some management functionalities to keep track of all interactions with the users by using logs and automatically generating annotated and anonymized XML files. In addition, the implementation also allows the chatbot to connect with an online service by following a simple protocol based on JSON messages and Websockets.

1 General Description

ELIZA¹ is a computer programs written by Joseph Weizenbaum between 1964 and 1966, and its main intention was to simulate a Rogerian psychotherapist [1]. Although it is based on simple pattern matching techniques using regular expressions and following a determined script, Eliza sometimes provides a startlingly human-like interaction and was taken seriously by several of its users at that time, even after being explained how it worked. In Weizenbaum's words, Eliza provided a "parody" of "the responses of a non-directional psychotherapist in an initial psychiatric

interview." [2], where by choosing this kind of "personality" the program did not require a database of real-world knowledge neither a deep understanding of the topic under discussion; therefore, Eliza usually replies to a user's statement with another question or reformulates the original question, for instance:

User: "Who is your favorite writer?"

Eliza's possible answers: "Does that question interest you?" or "What about your own favorite writer?"

2 Technical Description

Weizenbaum's implementation was done originally in MAD-SLIP list-processing language, therefore Eliza worked by simple parsing and substitution of key words into predefined sentences. Then, it has been ported to many other languages including Python by Jez Higgins and Joe Strouts². Our implementation is based on their version but updated to work on Python 2.7/3.4 and tested on Windows, Mac, and Linux.



Figure 1: left) Image for one of the online versions of Eliza³, and right) image selected for pyEliza

¹ <https://en.wikipedia.org/wiki/ELIZA>

² <http://www.jezuk.co.uk/cgi-bin/view/software/eliza>

³ <http://nlp-addiction.com/eliza/>

3 Sample Dialogue Sessions

Table 1 shows an example of an interaction with pyEliza. Since this implementation is not web-based but command-line based the user had to install the program from the pypi repository⁴ and running it with the following commands:

Install: pip install pyElizaChatbotClient --upgrade
 Run: python -m pyElizaChatbotClient -a

Table 1: Sample Session with PyEliza

1	pyEliza: Hello, my name is Eliza. Please tell me your name. User: Hi, my name is John
2	pyEliza: I am not sure if we have already met John, but anyway... What's up? User: No, this is the first time I meet you. I'm fine. What about you?
3	pyEliza: Please consider whether you can answer your own question. User: I'm fine
4	pyEliza: How does being fine make you feel? User: I feel good I guess. Some days better than others but in general happy
5	pyEliza: Do you often feel good you guess. Some days better than others but in general happy? User: Yes, today for instance I have good news at work
6	pyEliza: OK, but can you elaborate a bit? User: yes, my boss congratulates me because one of our clients accepted our offer
7	pyEliza: OK, but can you elaborate a bit? User: I have been contacting them for a while and pursuing them to change to our company as investor advisers
8	pyEliza: Why do you tell me that you've been contacting them for a while and pursuing them to change to our company as investor advisers? User: because, after all this work, finally they accepted and now we will have a good client in our portfolio
9	pyEliza: How does that make you feel? User: very happy and proud
10	pyEliza: Very interesting User: Yes. May be this could mean that I will get a promotion too
11	pyEliza: You seem quite sure. User: Not really, but this is a good success for all of us
12	pyEliza: Very interesting. User: Thanks. Ok. I need to leave. Bye

⁴ <https://pypi.python.org/pypi/pyElizaChatbotClient>

4 Additional capabilities

The current implementation easily allows programmers to modify the set of default answers and regular expressions (see the example below) to allow new interactions. In addition, it includes a function to automatically extract the user's name requested in the first turn.

```
[r'Can I ([^\?]* )??', # Regular expression
 [ "Perhaps you don't want to %1.", # Answers
  "Do you want to be able to %1?",
  "If you could %1, would you?"]],
```

On the other hand, although it is not directly related with the capabilities of the chatbot, our implementation includes: a) the possibility of connecting it as a client to an online platform⁵ by using Websockets and JSON messages, b) saving log files to keep track about all the interactions and debugging messages generated by the client when interacting with the users, and c) the possibility of generating XML files with anonymized turns to allow its posterior annotations and evaluation.

5 Future Work

We plan to extend the chatbot by a) including the possibility of increasingly learning new answers at each dialog turn as done by cleverbot⁶, b) include a history mechanism that will allow the chatbot to answer based on the current and most recent turns, and c) by including a new mechanism based on generative models to allow the creation of new answers instead of the predefined ones following a similar approach as proposed in [3].

References

- [1]. Weizenbaum, Joseph. "ELIZA—a computer program for the study of natural language communication between man and machine." Communications of the ACM 9.1 (1966): 36-45.,doi:10.1145/365153.365168
- [2]. Weizenbaum, Joseph. "Computer power and human reason: From judgment to calculation." (1976).
- [3]. Vinyals, Oriol, and Quoc Le. "A neural conversational model." arXiv preprint arXiv:1506.05869 (2015).

⁵ <http://www.teachabot.com:8000/>

⁶ <http://www.cleverbot.com/>

TickTock

RE-WOCHAT 2016 Shared Task Chatbot Description Report

Zhou Yu, Ziyu Xu, Alan W Black, Alexander I. Rudnicky

Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, 15217
{zhouyu, air, awb}@cs.cmu.edu, ziyux@andrew.cmu.edu

Abstract

This is a description of the TickTock chatbot system, which is a retrieval based system that utilizes conversational strategies to improve the system performance. It has two versions, one with multimodal signals as input; one with text input through typing. The multimodal version is a stand alone system (Yu et al., 2015), while the text version is a web-API version. In this report, we focus on describing the web-API version of TickTock, which is used in the shared task.

1. General Description

TickTock is a system that is capable of conducting free-form conversations, in contrast to goal-driven systems, which are designed to acquire information, provide feedback, or negotiate constraints with the human. A free-conversation system in principle removes any built-in value for the human and its success depends on the machine keeping the human interested in the ongoing conversation. Thus, as task completion is no longer an applicable metric, we chose to focus on metrics of the user’s experience, such as engagement, likability, and willingness to future interaction along with the appropriateness of the system responses. Similar to (Banchs and Li, 2012), TickTock is a retrieval based non-goal oriented dialog system. It generates the response by utilizing the corpus, which is a interview corpus. Different from other retrieval systems, it has strategies that handle situations where the retrieval methods could not produce appropriate responses.

2. TickTock System Description

TickTock is an retrieval based system with conversation strategy facilitation. A multimodal version of TickTock is described in (Yu et al., 2015), with similar architecture but with minor adjustments to the conversational strategies.

TickTock has a database that consists of question-answer pairs from CNN Interview Transcripts from the “Piers Morgan Tonight” show. The corpus has 767 Interviews in total and each interview is between 500 to 1,000 sentences. To construct our database, we used a rule-based question identification method, which simply means searching for tokens such as ‘?’, ‘How’, ‘Wh-’, etc. to identify questions and then extracted the consecutive utterance of the other speaker as the answer to that question. In total we have 67,834 pairs of utterances. Later we recruited users in Mechanical Turk to generate targeted utterances for this specific task (Yu et al., 2016).

Key Term Matching (Martin, 2002) was used for response generation. The user’s text input is first processed by the NLU component, which performs POS tagging and removes stop words; heuristics are then used to compute the database and calculate the weighted sum, which becomes the retrieval confidence score. Finally, we normalize the

score by dividing it by the length of the retrieved utterance. We filter out inappropriate content, excluding the retrieved answer if it is longer than 15 words and remove other characters such as parentheses or square brackets (along with everything between them). Our goal is to generate coherent conversations without deep understanding of the context, which is useful in a non-task oriented interactive system, and is motivated by lexical cohesion in modeling discourse. The coherence can be reflected by the repetition of lexicon items. The method first does shallow syntactic analysis of the input utterance and extracts keywords. These are used to search the corpus for a suitable response. We used six strategies described below to address some of the issues within the user input.

- 1. Single-word Sentence:** We found that some users were typing in meaningless single words such as ‘d’, ‘dd’, or equations such as ‘1+2=’. TickTock will reply ‘Can you be serious and say things in a complete sentence?’. We have a set of surface realization of such replies to choose from, so users would get a lightly different version every time, with the aim of making TickTock seem less robotic.
- 2. Out of Vocabulary:** We found that typos occur in the users’ responses and they used words that are not in the vocabulary of our database, such as ‘confrontational’. We implemented a strategy that when a sentence contains an out of vocabulary word, TickTock will reply with a clarification question, such as ‘What is ‘confrontational’?’ to communicate that it cannot understand his utterance entirely.
- 3. Anaphora:** We found user inputs with very limited concrete information in themselves, but referred to a prior response in the conversation. An example input would be “I hate them” and it is referring back to the ‘sports’ topic in the previous phrase, “How about we talk about sports?”. Anaphora is a difficult problem to solve for complex sentence structures. However in colloquial sentences, substituting in the noun of the previous sentence covers 85% of the cases. We implemented this simple rule to tackle anaphora.

Participant	TickTock(Appropriateness)
Hello how are you?	I'm feeling quite good. (3)
Anything new going on with you?	Of course. What do you think? (3)
I am not sure. What's on your mind?	Well, you know, I'm not sure. let me tell you something, Politicians and diapers have one thing in common. They should both be changed regularly, and for the same reason. (3)
Wow. That funny and insightful.	Uhh, I can't answer this. Can we talk about something else? (2)
How about movies. Have you seen any good movies?	Of course. What do you think? (2)
What have you seen?	Yes. (2)
I love super hero movies. Have you seen Deadpool?	Uh-huh, I do. (2)

Table 1: An example conversation with TickTock

- 4. Query Knowledge Base for Named Entities** A lot of Turkers assumed TickTock could answer factual questions, so they asked questions such as “Which state is Chicago in?”. We used the Wikipedia knowledge base API to answer such questions. We first performed a shallow parsing to find the named entity in the sentence, which we then searched for in the knowledge base, and retrieved the corresponding short description of that named entity. We then designed a template to generate sentences using the obtained short description of the mentioned name entity, such as “Are you talking about the city in Illinois?”.
- 5. Weight Adjustment with TF-IDF** We re-weighted the importance of the key words in an utterance based on its tf-idf score. Using POS tagging of the words that match between a user input, and the sentence a response is in reply to, we give nouns a score of 3, verbs a score of 2, and other words a score of 1. We then multiply each of these scores by the tf-idf value of the corresponding words, and the sum of their scores gives us the score of the response.
- 6. Incorporating One-utterance History** In ranking the retrieved response, we Incorporated the previous one turn context of the conversation. We compute the cosine similarity of the highly ranked response with the previous utterance, and picked the one that is more similar. We convert the utterances to vector space using word2vec method.

Once we retrieved the response, we select a conversational strategy, based on a heuristic, i.e. a predefined threshold for the retrieval confidence score, which can be tuned to make the system appear more active or more passive. Higher thresholds correspond to more active user engagement. When the retrieval confidence score is high, we return the found response in the database back to the user. If the retrieval confidence score is low, the dialog manager will choose a strategy that takes context into consideration. There are five strategies we used to deal with possible breakdowns the low retrieval confidence score indicates:

- 1. Switch topics:** propose a new topic other than the current topic, such as “sports” or “music”.
- 2. Initiate things to do:** propose something to do together, such as “Do you want to see the latest star war movie together?”.

- 3. End topics with an open question:** close the current topic using an open question, such as “Could you tell me something interesting?”.
- 4. Tell a joke:** tell a joke such as: “ Politicians and diapers have one thing in common. They should both be changed regularly, and for the same reason”.
- 5. Elicit more information:** ask the user to say more about the current topic, using utterances such as “ Could we talk more about that?â”.

3. An Example Conversation

In Table 1, we show an example conversation that TickTock produced.

4. Future Work

Our intent is to go beyond the response appropriateness and put more emphasis on overall discourse cohesion. For example, there is a breakdown type we have not addressed, which is the chatbot’s inconsistency in adhering to the context of the conversation. A possible solution would be to maintain a knowledge base of what the user said and use it for consistency checking as part of the selection process for the final response.

We are also interested in determining how the system can channel a conversation into a specific topic. That is, if TickTock starts the conversation with a given topic, how long and with what strategies will it be able to keep the user on the same topic. We also wish to develop strategies that elicit high quality responses from human users (perhaps as a consequence of maintaining a high level of engagement).

5. References

- Banchs, R. E. and Li, H. (2012). Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the ACL 2012 System Demonstrations*, pages 37–42. Association for Computational Linguistics.
- Martin, J. R. (2002). *Meaning beyond the clause: area: self-perspectives*. Annual Review of Applied Linguistics 22.
- Yu, Z., Papangelis, A., and Rudnicky, A. (2015). TickTock: A non-goal-oriented multimodal dialog system with engagement awareness. In *Proceedings of the AAAI Spring Symposium*.
- Yu, Z., Xu, Z., Black, A., and Rudnicky, A. (2016). Chatbot evaluation and database expansion via crowdsourcing. In *Proceedings of the chatbot workshop of LREC*.

Sarah Chatbot

RE-WOCHAT 2016 - SHARED TASK

CHATBOT DESCRIPTION REPORT

Bayan AbuShawar
IT department;
Arab Open University
Amman, Jordan
b shawar@aoou.edu.jo

Abstract

Sarah chatbot is a prototype of ALICE chatbot, using the same knowledge base (AIML) files of ALICE. Sarah was created to enable public chatting with it using the pandorobot host serving. The Loebner prize¹ competition has been used to evaluate machine conversation chatbots. The Loebner Prize is a Turing test, which evaluates the ability of the machine to fool people that they are talking to human. In essence, judges are allowed a short chat (10 to 15 minutes) with each chatbot, and asked to rank them in terms of “naturalness”.

1 ALICE General Description

A.L.I.C.E.² is the Artificial Linguistic Internet Computer Entity, which was implemented by Wallace in 1995. Alice knowledge about English conversation patterns is stored in AIML files. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML). It was developed by Wallace and the Alicebot free software community during 1995-2000 to enable people to input dialogue pattern knowledge into chatbots based on the A.L.I.C.E. open-source software technology.

¹ <http://www.loebner.net/Prizef/loebner-prize.html>

² <http://www.Alicebot.org/>

2 ALICE Technical Description

AIML consists of data objects called AIML objects, which are made up of units called topics and categories. The topic is an optional top-level element, has a name attribute and a set of categories related to that topic. Categories are the basic unit of knowledge in AIML. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which matches against the user input, and a template, which is used in generating the ALICE chatbot answer, the format of AIML is as follows:

```
<category>  
<pattern>PATTERN</pattern>  
<template>Template</template>  
</category>
```

The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols `_` and `*`. The words may consist of letters and numerals, but no other characters. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant. The idea of the pattern matching technique is based on finding the best, longest, pattern match.

3 Sample Dialogue Sessions

Table 1 represents a sample of chatting with Sarah bot.

1	USER: good morning
2	SarahBot: Hello and how are you this morning?
3	USER: I am very tired
4	SarahBot: Better get some rest.
5	USER: no, I caanot, I have a lot of work to do
6	SarahBot: I have no idea what to say.
7	USER: you do not need to say anything
8	SarahBot: Do humans need to do it?
9	USER: do wat?
10	SarahBot: I'm sorry, unknown, I don't have enough information.
11	USER: do wat?
12	SarahBot: Do whatever you want.
13	USER: I would like to do whatever I want, but regrettably I need to work
14	SarahBot: I want to become smarter than humans.

4 The Pandorobot Host Services

The pandorobot³ web-hosting service was used to publish different prototypes of ALICE, where the user can create his own chatbot, upload his own dialogue or used the basic AIML files of ALICE in

³ <http://www.pandorabots.com/pandora>

addition to his own ones. Pandorabots.com, hosts thousands of Echatbots built using the AIML format. The most popular Pandorabots for the last 24 hours web-page regularly lists chatbots developed by researchers and hobbyists, and also some commercial systems. For example, Cyber-Sandy and Nickie act as portals to adult-entertainment websites; Jenny introduces the English2Go website, and lets English language learners practice their chatting technique.

5 The Loebner Prize Competition

The story began with the “imitation game” which was presented in Alan Turing’s paper Can Machine think?. The imitation game has a human observer who tries to guess the sex of two players, one of which is a man and the other is a woman, but while screened from being able to tell which is which by voice, or appearance. Turing suggested putting a machine in the place of one of the humans and essentially playing the same game. If the observer cannot tell which is the machine and which is the human, this can be taken as strong evidence that the machine can think.

Turing’s proposal provided the inspiration for the Loebner Prize competition, which was an attempt to implement the Turing test. The first open-ended implementation of the Turing Test was applied in the 1995 contest, and the prize was granted to Weintraub for the fourth time. For more details to see other winners over years are found in the Loebner Webpage.