The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems

Ryan Lowe*, Nissan Pow*, Iulian Serban†, Joelle Pineau*

*McGill University
†Université de Montréal

June 16, 2015

Overview

1. Dialogue Datasets
   - The Ubuntu Dialogue Corpus
   - Evaluation Metrics
2. Implemented Algorithms
   - Neural Models
   - TF-IDF Baseline
3. Future Work

Ubuntu Chat Corpus

Contains several years of chat logs, with the following characteristics:

- Millions of utterances
- Multi-party (however we can extract dialogues)
- Application towards technical support

Example Conversation

[12:21] greg: have people had problems using automatix? specifically firefox
[12:21] sybariten: amphi: ok, i'm trying to set IRSSI to get the character "emulation" ISO-8859-1 ... aka "western"
[12:21] ruchbah: sybariten .. nope. No error.
[12:21] gnomefreak: greg: dont use it
[12:21] sybariten: ruchbah: ok, then it works for you ... dang

Dialogue Extraction Method

Use the fact that users **specifically address** the users they are talking to.

- Identify utterances where two users address each other.
- Work backwards to find the **original question** of first user.
- If users only address each-other in this time, include all utterances from both users.
- Discard dialogues where one user has >80% of the utterances, and merge consecutive utterances by same user.

Dialogue Extraction Method: Example

Time	User	Utterance
03:44	Old	I dont run graphical ubuntu, I run ubuntu server.
03:45	kuja	Taru: Haha sucker.
03:45	Taru	Kuja: ?
03:45	bur[n]er	Old: you can use ”ps ax” and ”kill (PID#)”
03:45	kuja	Taru: Anyways, you made the changes right?
03:45	Taru	Kuja: Yes.
03:45	LiveCD	or killall speedlink
03:45	kuja	Taru: Then from the terminal type: sudo apt-get update
03:46	_pm	if i install the beta version, how can i update it when the final version comes out?
03:46	Taru	Kuja: I did.

Sender	Recipient	Utterance
Old		I dont run graphical ubuntu, I run ubuntu server.
bur[n]er	Old	you can use ”ps ax” and ”kill (PID#)”
kuja	Taru	Haha sucker.
Taru	Kuja	?
kuja	Taru	Anyways, you made the changes right?
Taru	Kuja	Yes.
kuja	Taru	Then from the terminal type: sudo apt-get update
Taru	Kuja	I did.

Figure: Example chat room conversation from the #ubuntu channel of the Ubuntu Chat Logs (left), with the disentangled conversations for the Ubuntu Dialogue Corpus (right).

Ubuntu Dataset Properties

There are about 1 million dialogues with 3 or more turns. Of these dialogues, the average number of turns is 8.

# dialogues (human-human)	932,429
# utterances (in total)	7,189,051
# words (in total)	100,000,000
Min. # turns per dialogue	3
Avg. # turns per dialogue	7.71
Avg. # words per utterance	10.34
Median conversation length (min)	6

Table: Properties of Ubuntu Dialogue Corpus.

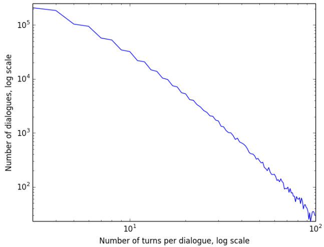

Figure: The distribution of the number of turns. Both axes are log scale.

Evaluation Metrics

How to determine if the dialogue model you are using is good?
Can use:

Slot filling, used in the Dialogue State Tracking Challenge.

- Limited in terms of the data available and generalization to other domains.

Prediction of the next utterance given previous context.

- Predicted sentences can be very reasonable, yet completely different from actual utterance.
- Use BLEU score from machine translation.

Evaluation Metrics

Can use 'multiple choice'-style questions, choosing most likely next utterance given a past context.

- Easier than generating a full response.
- Can adjust problem difficulty.
- **Idea:** Any model that can *generate* 'good' dialogue, should be able to *recognize* 'good' dialogue.

Context	Response	Flag
well, can I move the drives? __EOS__ ah not like that	I guess I could just get an enclosure and copy via USB	1
well, can I move the drives? __EOS__ ah not like that	you can use "ps ax" and "kill (PID #)"	0

Table: To train the model, use (context, response, flag) triples.

Aside: Word Embeddings

When training the RNN, represent each word as a vector in an **embedded feature space**:

- Can be pre-trained, or done jointly with the language model.
- Pre-trained vectors (GloVe or `word2vec`) computed using the **distributional similarity** of surrounding words.
- We initialize using GloVe, and fine-tune using dialogue data.

Recurrent Neural Networks (RNNs)

- Variant of neural nets that allow for **directed cycles** between units.
- Leads to **hidden state** of the network, h_t, which allows it to model time-dependent data.

$$h_t = f(h_{t-1}, x_t)$$

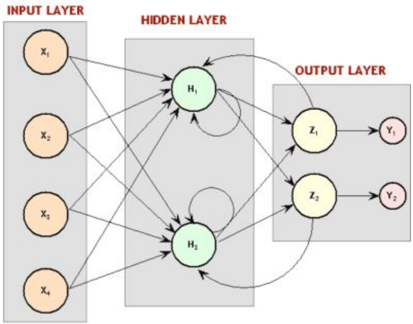

Figure: Image source: `www.deeplearning.net`

Long-Short Term Memory (LSTMs)

- Introduces gating mechanism to RNNs.
- Improves on the long-term memory capabilities of RNNs.
- Primary building block of many current neural language models.

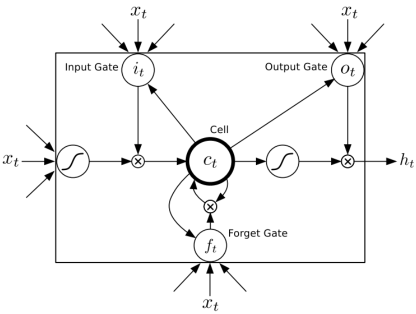

Figure: Image source: Graves (2014)

Neural Dialogue Model

- First calculate embeddings of context/reply with RNNs.
- Probability of the given reply being the actual reply is then:

$$p(flag = 1|\boldsymbol{c}, \boldsymbol{r}) = \sigma(\boldsymbol{c}^T \boldsymbol{M}\boldsymbol{r} + b)$$

  where b is a bias term and $\boldsymbol{M}$ are learned parameters.
- Can be thought of as the dot product between $\boldsymbol{c}$ and some generated context $\boldsymbol{Mr}$.

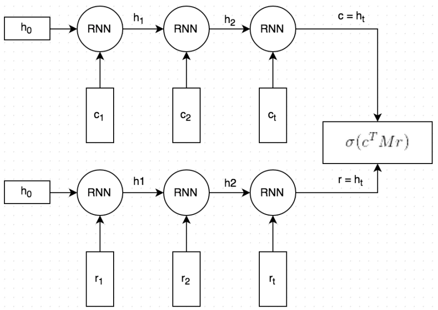

Figure: Diagram of the model. c_i are word vectors for the *context* (top), r_i for the *response* (bottom).

Neural Dialogue Model

- Model's RNNs have tied weights.
- We consider contexts up to a maximum of $t = 160$.
- Model is trained by minimizing the cross-entropy of context/reply pairs:

$$\mathcal{L} = -\log \prod_n p(flag_n | \mathbf{c_n}, \mathbf{r_n}) + \frac{\lambda}{2} ||\theta||_2^F$$

- Adapted from the approach in Bordes et al. (2014) and Yu et al., (2014) for question answering.

Term Frequency - Inverse Document Frequency

- Captures how important a given word is to some document.
- We calculate TF-IDF score for each word in each candidate reply. Reply with **highest average score** is selected.

- Calculated using:

$$\text{tfidf}(w, c, C) = f(w, c) \times \log \frac{N}{|\{c \in C : w \in c\}|}$$

where $f(w, c)$ is # of times word w appeared in context C, N is total # of dialogues, denominator represents the # of dialogues with w.

Results

Method	TF-IDF	RNN	LSTM
1 in 2 R@1	65.9%	74.4%	**87.7%**
1 in 10 R@1	41.0%	36.9%	**60.2%**
1 in 10 R@2	54.5%	50.4%	**74.6%**
1 in 10 R@5	70.8%	79.0%	**92.7%**

Table: Results for the three algorithms using various recall measures for binary (1 in 2) and 1 in 10 (1 in 10) next utterance classification %, using 1/8th of the data.

Effect of Dataset Size

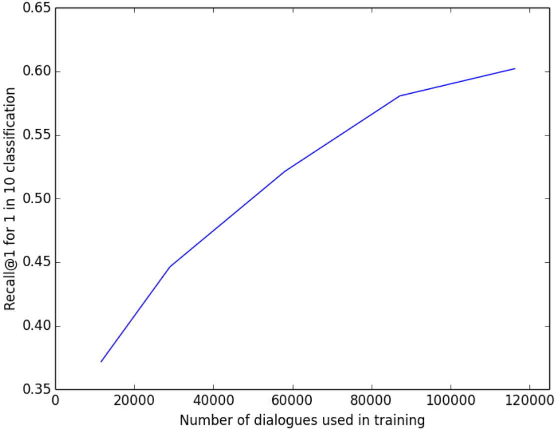

Figure: The LSTM (with 200 hidden units), showing Recall@1 for the 1 in 10 classification, with increasing dataset sizes.

Future Work

Ensuring the quality of the final dataset:

- Perform **human trials**.
- Experiment with other chat disentanglement methods

Improving architectures for modeling dialogues:

- Investigate other neural architectures.
- Experiment with attention over the context.
- Investigate methods of finding embeddings for out-of-vocabulary (OOV) words.
- Incorporate external domain-specific knowledge.

References

A. Bordes, J. Weston, and N. Usunier.
Open question answering with weakly supervised embedding models.
In *MLKDD*, pages 165–180. Springer, 2014.

K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio.
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
arXiv preprint arXiv:1406.1078, 2014.

S. Hochreiter and J. Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.

A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.Y. Nie, J. Gao, and W. Dolan.
A neural network approach to context-sensitive generation of conversational responses.
2015.

C.C. Uthus and D.W Aha.
Extending word highlighting in multiparticipant chat.
Technical report, DTIC Document, 2013.

L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman.
Deep learning for answer sentence selection.
arXiv preprint arXiv:1412.1632, 2014.

Questions?