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] gnomefreak: greg: dont use it
[12:21] sybariten: ruchbah: ok, then it works for you ... dang
Use the fact that users \textit{specifically address} the users they are talking to.

- Identify utterances where two users address each other.
- Work backwards to find the \textit{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

**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).

<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>03:44</td>
<td>Old</td>
<td>I don't run graphical ubuntu, I run ubuntu server.</td>
</tr>
<tr>
<td>03:45</td>
<td>kuja</td>
<td>Taru: Haha sucker.</td>
</tr>
<tr>
<td>03:45</td>
<td>Taru</td>
<td>Kuja: ?</td>
</tr>
<tr>
<td>03:45</td>
<td>bur[n]er</td>
<td>Old: you can use &quot;ps ax&quot; and &quot;kill (PID#)&quot;</td>
</tr>
<tr>
<td>03:45</td>
<td>kuja</td>
<td>Taru: Anyways, you made the changes right?</td>
</tr>
<tr>
<td>03:45</td>
<td>Taru</td>
<td>Kuja: Yes.</td>
</tr>
<tr>
<td>03:45</td>
<td>LiveCD</td>
<td>or killall speedlink</td>
</tr>
<tr>
<td>03:45</td>
<td>kuja</td>
<td>Taru: Then from the terminal type: sudo apt-get update</td>
</tr>
<tr>
<td>03:46</td>
<td>-pm</td>
<td>if i install the beta version, how can i update it when the final version comes out?</td>
</tr>
<tr>
<td>03:46</td>
<td>Taru</td>
<td>Kuja: I did.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sender</th>
<th>Recipient</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
<td></td>
<td>I don't run graphical ubuntu, I run ubuntu server.</td>
</tr>
<tr>
<td>bur[n]er</td>
<td>Old</td>
<td>you can use &quot;ps ax&quot; and &quot;kill (PID#)&quot;</td>
</tr>
<tr>
<td>kuja</td>
<td>Taru</td>
<td>Haha sucker.</td>
</tr>
<tr>
<td>Taru</td>
<td>Kuja</td>
<td>?</td>
</tr>
<tr>
<td>kuja</td>
<td>Taru</td>
<td>Anyways, you made the changes right?</td>
</tr>
<tr>
<td>Taru</td>
<td>Kuja</td>
<td>Yes.</td>
</tr>
<tr>
<td>kuja</td>
<td>Taru</td>
<td>Then from the terminal type: sudo apt-get update</td>
</tr>
<tr>
<td>Taru</td>
<td>Kuja</td>
<td>I did.</td>
</tr>
</tbody>
</table>

Ryan Lowe (McGill University)  
Samsung Workshop  
June 16, 2015
There are about 1 million dialogues with 3 or more turns. Of these dialogues, the average number of turns is 8.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># dialogues (human-human)</td>
<td>932,429</td>
</tr>
<tr>
<td># utterances (in total)</td>
<td>7,189,051</td>
</tr>
<tr>
<td># words (in total)</td>
<td>100,000,000</td>
</tr>
<tr>
<td>Min. # turns per dialogue</td>
<td>3</td>
</tr>
<tr>
<td>Avg. # turns per dialogue</td>
<td>7.71</td>
</tr>
<tr>
<td>Avg. # words per utterance</td>
<td>10.34</td>
</tr>
<tr>
<td>Median conversation length (min)</td>
<td>6</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Context</th>
<th>Response</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>well, can I move the drives?</td>
<td>I guess I could just get an enclosure and copy via USB</td>
<td>1</td>
</tr>
<tr>
<td><em>EOS</em> ah not like that</td>
<td></td>
<td></td>
</tr>
<tr>
<td>well, can I move the drives?</td>
<td>you can use ”ps ax” and ”kill (PID #)”</td>
<td>0</td>
</tr>
<tr>
<td><em>EOS</em> ah not like that</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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.
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:

\[ p(\text{flag} = 1 | \textbf{c}, \textbf{r}) = \sigma(\textbf{c}^T \textbf{Mr} + b) \]

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

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

$$\mathcal{L} = -\log \prod_{n} p(flag_n|c_n, r_n) + \frac{\lambda}{2} ||\theta||^2_F$$

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:

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

where \(f(w, c)\) is \# of times word \(w\) appeared in context \(C\), \(N\) is total \# of dialogues, denominator represents the \# of dialogues with \(w\).
<table>
<thead>
<tr>
<th>Method</th>
<th>TF-IDF</th>
<th>RNN</th>
<th>LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 in 2 R@1</td>
<td>65.9%</td>
<td>74.4%</td>
<td>87.7%</td>
</tr>
<tr>
<td>1 in 10 R@1</td>
<td>41.0%</td>
<td>36.9%</td>
<td>60.2%</td>
</tr>
<tr>
<td>1 in 10 R@2</td>
<td>54.5%</td>
<td>50.4%</td>
<td>74.6%</td>
</tr>
<tr>
<td>1 in 10 R@5</td>
<td>70.8%</td>
<td>79.0%</td>
<td>92.7%</td>
</tr>
</tbody>
</table>

**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.
A. Bordes, J. Weston, and N. Usunier.
Open question answering with weakly supervised embedding models.

Learning phrase representations using rnn encoder-decoder for statistical machine translation.

S. Hochreiter and J. Schmidhuber.
Long short-term memory.

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.

Deep learning for answer sentence selection.
Questions?