Training a Chatbot with Microsoft LUIS: Effect of Intent Imbalance on Prediction Accuracy

Ruane, Elayne; Young, Robert; Ventresque, Anthony

2020-03-17

IUI '20: Proceedings of the 25th International Conference on Intelligent User Interfaces Companion

The 25th International Conference on Intelligent User Interfaces Companion (IUI'20), Cagliari, Italy, 17-20 March 2020

ACM

http://iui.acm.org/2020/

http://hdl.handle.net/10197/11782

© the Authors, 2020. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in IUI '20: Proceedings of the 25th International Conference on Intelligent User Interfaces Companion https://doi.org/10.1145/3379336.3381494

10.1145/3379336.3381494

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information, please see the item record link above.
Training a Chatbot with Microsoft LUIS: Effect of Intent Imbalance on Prediction Accuracy

Elayne Ruane  
elayne.ruane@ucdconnect.ie  
University College Dublin, Ireland  
Lero - The Irish Software Research Centre

Robert Young  
robert.young2@ucdconnect.ie  
University College Dublin, Ireland  
Lero - The Irish Software Research Centre

Anthony Ventresque  
anthony.ventresque@ucd.ie  
University College Dublin, Ireland  
Lero - The Irish Software Research Centre

ABSTRACT

Microsoft’s LUIS is a cloud-based language understanding API that uses Machine Learning (ML) to predict sentence meaning and extract information for training chatbots. To develop a chatbot app using LUIS we define a number of intents, an abstraction of a task the user may want to do using the chatbot. LUIS is a supervised learning model requiring labelled example utterances to train an intent. For example, we can define an intent Order Taxi and create utterances such as "please book me a taxi now".

When a natural language utterance is submitted to the chatbot, the underlying LUIS model will parse it and try to classify it. LUIS will return the top intent along with a confidence score. Due to the nature of natural language, some intents will have more variability in how they may be expressed by the user than others. For example, a Greeting intent may have 30 example utterances but a None intent should contain examples that fall outside the chatbot domain and comprise 10% of all training utterances. LUIS uses non-deterministic training so if two intents are trained on very similar utterances and have similar scores for an utterance, the top intent may invert and become the second-top intent. It is recommended to have a 15% difference in scores to avoid this. As such, we want to train our chatbot such that the top intent for recognised utterances has both a high confidence score and a significant margin between the top intent and the rest.

We used two existing datasets to train three LUIS apps. The first dataset is a large multi-topic crowdsourced dataset with 7 task-based intents. The dataset is unbalanced but the None intent has the majority class. The black-box nature of LUIS means examples are no example utterances for a None intent. The LUIS docs discuss best practice for designing intents [1]. It is recommended to use 15 to 30 specific and varying example utterances for each intent, only adding further examples after training and testing. A None intent should contain examples that fall outside the chatbot domain and comprise of all training utterances. LUIS uses non-deterministic training so if two intents are trained on very similar utterances and have similar scores for an utterance, the top intent may invert and become the second-top intent. It is recommended to have a 15% difference in scores to avoid this. As such, we want to train our chatbot such that the top intent for recognised utterances has both a high confidence score and a significant margin between the top intent and the rest.

2 RELATED WORK

The LUIS docs discuss best practice for designing intents [1]. It is recommended to use 15 to 30 specific and varying example utterances for each intent, only adding further examples after training and testing. A None intent should contain examples that fall outside the chatbot domain and comprise of all training utterances. LUIS uses non-deterministic training so if two intents are trained on very similar utterances and have similar scores for an utterance, the top intent may invert and become the second-top intent. It is recommended to have a 15% difference in scores to avoid this. As such, we want to train our chatbot such that the top intent for recognised utterances has both a high confidence score and a significant margin between the top intent and the rest.

Much work has been done on dataset imbalance for ML models. Mirończuk and Protasiewicz (2019) [5] provide a detailed survey of text classification literature including work focusing on the development of models that can handle class imbalance. Other studies look at how to address the problem by processing the dataset itself using techniques such as under- and over-sampling [4]. However, re-sampling methods are not feasible here due to the small data set size and the redundancy of duplicate examples.

The class imbalance problem often occurs in binary classification or anomaly detection scenarios that try to identify a particular event that occurs much less frequently than the majority class e.g. insurance fraud detection. In our context, we have a greater number of classes (intents) with less structured data (natural language). As such, the imbalance will affect unseen utterances differently depending on their structure and content, unlike more clear-cut problems where the issue presents itself solely as prediction towards the majority class. The black-box nature of LUIS means examples are provided as strings and any pre-processing tasks such as tokenisation, stop-word removal, and lemmatisation, are all done "under the hood". Document representation, feature selection, and even the parameters of the model itself are unknown to us. As such, we focus solely on the training utterances.

3 DATASETS

We used two existing datasets to train three LUIS apps. The first dataset, "AskUbuntu", is a question and answer dataset scraped from askubuntu.com [2] containing questions seeking technical support. The dataset is unbalanced but the None intent has the recommended amount of example utterances. The second dataset is a large multi-topic crowdsourced dataset with 7 task-based intents. We trained an app MultiTask, using all of the example utterances in the dataset. The 7 task intents are balanced but there are no example utterances for a None intent.
Table 1: Dataset Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Intents</th>
<th>Utterances</th>
<th>Balanced</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AskUbuntu</td>
<td>5</td>
<td>162</td>
<td>No</td>
<td>100%</td>
</tr>
<tr>
<td>MultiTask</td>
<td>8</td>
<td>13,585</td>
<td>Yes</td>
<td>99.5%</td>
</tr>
<tr>
<td>MultiSmall</td>
<td>9</td>
<td>114</td>
<td>Yes</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 2: Batch Test Results (25 utt.) - AskUbuntu

<table>
<thead>
<tr>
<th>Intent</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
<th>Utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>MakeUpdate</td>
<td>1</td>
<td>0.6</td>
<td>0.75</td>
<td>47</td>
</tr>
<tr>
<td>None</td>
<td>N/A</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>SetupPrinter</td>
<td>1</td>
<td>0.8</td>
<td>0.89</td>
<td>23</td>
</tr>
<tr>
<td>ShutdownComp</td>
<td>1</td>
<td>0.8</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>SoftwareRec</td>
<td>0.38</td>
<td>1</td>
<td>0.55</td>
<td>57</td>
</tr>
</tbody>
</table>

We randomly selected 15 example utterances for each of the 7 task intents and trained a third app, MultiSmall, and manually added example utterances for a None intent to produce a dataset that followed all recommendations.

4 RESULTS AND DISCUSSION

Using the LUIS batch testing feature, we tested each app on 5 unseen utterances per intent. The results for the AskUbuntu dataset (Table 2) appear to be affected by intent imbalance. Majority intent SoftwareRec had perfect recall but very low precision due to a high number of false positives (Figure 1), including all five test utterances for the None intent. MultiSmall showed good performance on all but one intent (BookRestaurant) due to two false positives that were very similar in structure to the example utterances. Even better results were found for MultiTask despite a lower F-score for the SearchMedia intent which was due to four false positives because the None intent had no example utterances.

We explored these results by altering the datasets and repeating batch testing. We balanced the AskUbuntu dataset by reducing the lowest scoring utterances of the larger example sets and manually adding utterances to the other intents. Precision for the SoftwareRec class increased from 0.38 to 0.45 when the dataset was balanced suggesting that intent imbalance is part of the problem. When the balanced dataset MultiSmall was imbalanced towards BookRestaurant with an additional 10 training utterances there was no difference to the scores of any intent. However, when the SearchMedia intent example set was increased to 100, we see its precision drop to 0.31 on the test set due to an increase number of false positives. Of course, this means the recall of the other intents also drops as they are incorrectly classified as SearchMedia.

These initial experiments suggest intent imbalance can lead to a decrease in chatbot quality due to prediction towards the majority class. Further work is needed to explore more nuanced effects of example utterance structure similarities across intents.

ACKNOWLEDGMENTS

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094

REFERENCES