Active Learning: Select Training Data for Conversational AI Systems NLU from Logs

Praneeth Gubbala
Aug 20, 2019 · 3 min read

Conversation AI / Virtual Assistant systems responses are dependent on the machine learning models trained on the labeled training data. Constantly retraining Virtual Assistant models on new real-life data improves their performance. However, annotating all or a random sample of data is expensive, slow and inefficient. So we would like to annotate only the most informative utterances to reduce the error rate of NLU (Natural Langauge Understanding) system by new active learning algorithm called Majority-EnsembleCRF: a modified version of Majority-CRF(Peshterliev et al., 2019).

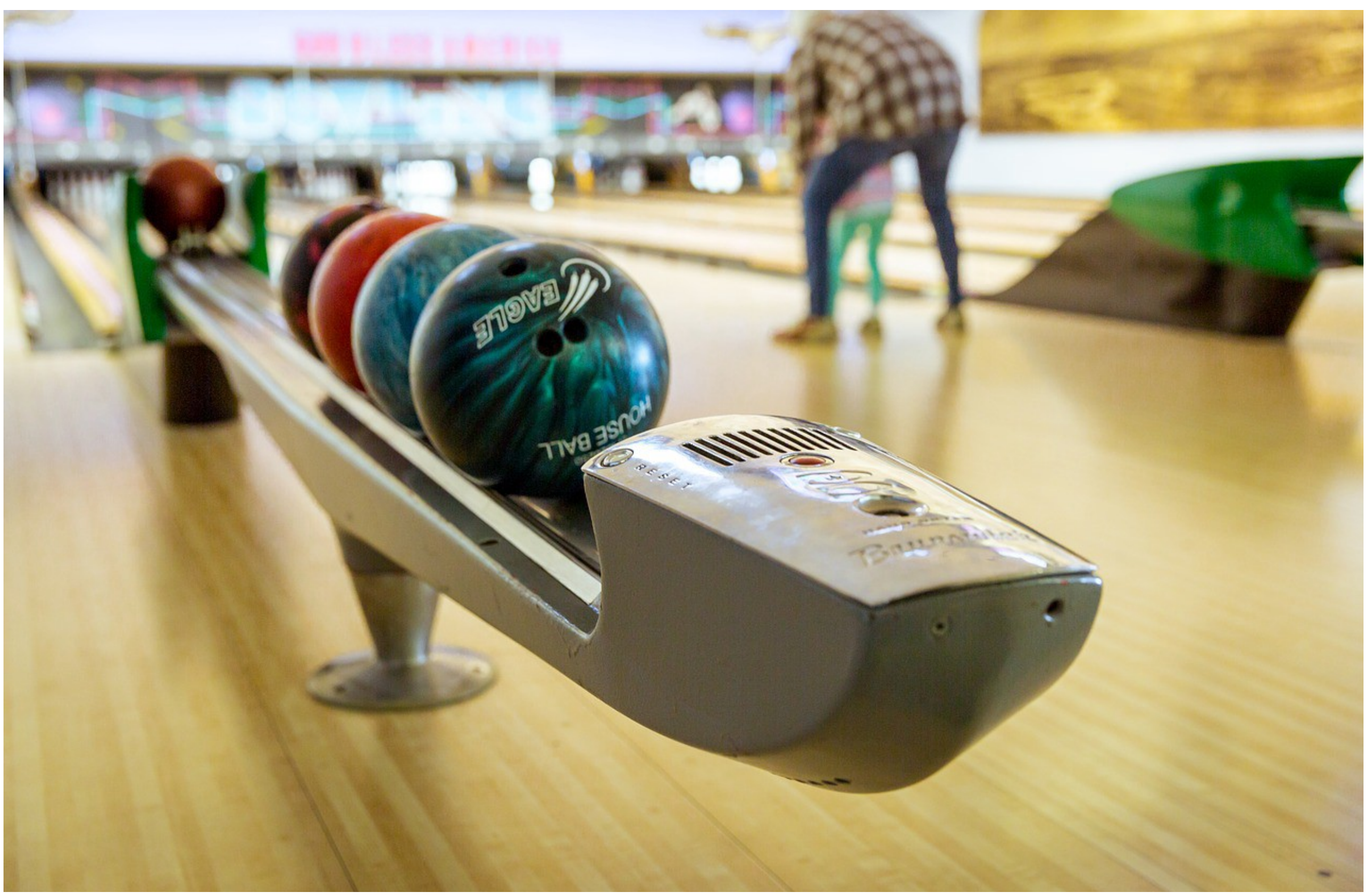

Photo credit: Active Learning

What is Active Learning?

Active Learning is a semi-supervised machine learning algorithm that chooses the training data it wants to learn from so that NLU can improve its accuracy with lesser data. It iteratively interacts with users to expand the NLU class training dataset and thus improve the prediction accuracy. This is a human-in-the-loop technique.

Text utterance examples with low confidence/ more uncertainty in Intent Classification models are important. Also, feeding/ training these models on how they should be labeled is informative. Higher uncertainty examples will give more information gain in the model. Selecting these examples automatically is called active learning.

Let’s see the Active Learning system for informative training data prediction for one NLU class/skill.

Implementation Details:

Pre-trained Models: First, you need to train your NLU system using the labeled data so it can classify intent and extract entities. Your Intent Classifier models can be based on an ensemble of models including Sklearn SVC (sklearn.svm), StarSpace: Embed All The Things! (StarSpace), BERT (BERT) and for entities, the seq-seq labeling models Conditional Random Field (CRF), BERT (BERT).

Preprocessing: From the Pool of unannotated live utterances, remove the utterances which are scored with high confidence (0.8) by the current NLU system. Now, the remaining Pool data is passed to the next step.

Modeling: Train the binary classifiers with different loss functions like a hinge, squared, logistic, etc with {1,2,3}-gram features. The training data for this model will be positive and negative data of the NLU skill/class. The classifiers implemented in Vowpal Wabbit (Langford et al., 2007). Once the set of models is ready, the next step is filtering.

Filtering: Extract only the majority positive prediction from the set of Intent Classifiers, into a filtered pool of utterances.

Scoring: For each pool utterance in the filtered pool, the score *s* is calculated as logistic classifier probability multiplied with NER system probability. Prioritize the utterances with the smallest score as more informative examples for a given NLU skill.

Manual Annotation: We prioritize by the smallest score *s* for the filtered pool utterances and deduplicate utterances before annotation. For a given min-batch size(k) for the informative training samples of NLU skill: the N informative utterances range from $(0 \leq N \leq k)$.

Repeat: Repeat this whole process of Active Learning until you get a good Conversion rate for the NLU Intent.

Impact areas: General Conversation AI systems.

References:

[1] Peshterliev, Kearney, Jagannatha, Kiss, Matsoukas (2019). Active Learning for New Domains in Natural Language Understanding. https://arxiv.org/abs/1810.03450v2

[2] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[3] Wu, Fisch, Chopra, Adams, Bordes, Weston. 2017. StarSpace: Embed All The Things!. https://arxiv.org/abs/1709.03856

[4] John Langford, Lihong Li, and Alex Strehl. 2007. Vowpal Wabbit

Machine Learning Data Science NLP Deep Learning Virtual Assistant

Download on the App Store

GET IT ON Google Play