An Active Learning Pipeline for NLU Error Detection in Conversational Agents

Damián Pascual, Aritz Bercher, Akansha Bhardwaj, Mingbo Cui, Dominic Kohler, Liam van der Poel, Paolo Rosso
Telepathy Labs GmbH
Zürich, Switzerland
{firstname.lastname}@telepathy.ai

Abstract

High-quality labeled data is paramount to the performance of modern machine learning models. However, annotating data is a time-consuming and costly process that requires human experts to examine large collections of raw data. For conversational agents in production settings with access to large amounts of user-agent conversations, the challenge is to decide what data should be annotated first. We consider the Natural Language Understanding (NLU) component of a conversational agent deployed in a real-world setup with limited resources. We present an active learning pipeline for offline detection of classification errors that leverages two strong classifiers. Then, we perform topic modeling on the potentially mis-classified samples to ease data analysis and to reveal error patterns. In our experiments, we show on a real-world dataset that by using our method to prioritize data annotation we reach 100% of the performance annotating only 36% of the data. Finally, we present an analysis of some of the error patterns revealed and argue that our pipeline is a valuable tool to detect critical errors and reduce the workload of annotators.

1 Introduction

Modern machine learning methods rely heavily on the availability of high-quality labeled data (Ouyang et al., 2022; Schuhmann et al., 2022). As a consequence, annotating large volumes of data has become a priority across organizations. However, data annotation is a time-consuming and costly process: it requires, first, to train human experts who, then, have to manually examine large collections of raw data and assign labels. Since assigning labels is often an ambiguous task, it is a standard that each sample is labeled by multiple annotators and labels are assigned based on inter-annotator agreement (Artstein, 2017). The complexity of this process makes data annotation a common bottleneck when it comes to deploying data-driven systems that should operate reliably in production environments.

A relevant example of these data-driven systems are conversational agents that interact directly with human users. These agents typically have at least two components, one for Natural Language Understanding (NLU) and another for Dialogue Management (DM). The NLU component extracts intents and entities from the user utterance at each conversation turn, while the DM component decides on the next action based on the NLU output (Bocklisch et al., 2017). Once deployed, these assistants can have access to large amounts of raw data in the form of user-agent conversations. At scale, the amount of data available for annotation may soon exceed the capacity of the human annotators. The challenge then becomes how to select samples for annotation. On the NLU side, it is desirable to prioritize the annotation of utterances whose intent was misclassified during inference in order to correct existing flaws in the agent. However, automatically finding those utterances is challenging, since intent mis-classifications do not necessarily result in failed conversations and conversations can fail due to the misbehavior of other components of the digital assistant, not only due to the NLU.

In this work, we consider a real-world scenario where an intent classifier needs to run with limited resources, specifically, in CPUs and with low latency. This discards modern Large Language Models (LLMs) as a valid option. Nevertheless, LLMs can be used offline to detect potentially mis-classified data. We present a simple yet effective method based on voting that leverages two LLMs to detect problematic utterances. In particular, we compare the prediction of two LLMs with the intent assigned by the production classifier and if there is no unanimity between the three intents, we mark the utterance as problematic to

Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII), pages 55–60

prioritize its annotation and analysis. We embed this method in an active learning pipeline consisting of error detection, clustering and topic modeling, followed by expert annotation. This way, the human expert receives a curated set of problematic utterances clustered by topic, which facilitates the discovery of error patterns and greatly reduces the required workload.

In our experiments, we simulate a real-world environment, where an intent classifier is periodically exposed to new data that can be potentially labeled and incorporated to the training data. We evaluate on a held-out test set and show that on a real-world dataset, an intent classification model trained with data labeled following the priority given by our pipeline can reach with 36% of the train data the same performance as with 100%, which represents a major reduction in annotation costs. Furthermore, we show a qualitative analysis of the error patterns discovered by our method on two public datasets and argue that our pipeline is a valuable tool to early-detect intent classification errors that could be critical for the operation of a conversational agent.

2 Related Work

Error Discovery: Error discovery strategies in machine learning can be categorised into machine-initiated (or, active learning) and human-initiated. While human-initiated approaches put a significant load on humans (Attenberg and Provost, 2010; Attenberg et al., 2015), machine initiated approaches are either based on dialogue failure (Khaziev et al., 2022), disagreement with the expectation (Bhardwaj et al., 2020, 2022), or confidence of the classifier (Lewis and Catlett, 1994). To label individual data instances, existing active learning strategies mainly leverage crowds (Yan et al., 2011; Yang et al., 2018) or components of a machine learning system (Nushi et al., 2017). Detecting feature blindness errors, namely unknown unknowns, with active learning methods is hard, since these methods generally rely on the model's training results (Attenberg et al., 2015; Lakkaraju et al., 2017). To mitigate this limitation, our error prediction workflow involves different machine learning models, diversifying in this way the type of errors discovered.

Interactive machine learning (iML): iML is a growing field in machine learning that has demonstrated its success in building well-performing classifiers using fewer features (Fails and Olsen Jr, 2003; Ware et al., 2001; Chen et al., 2018). Moreover, it improves user's trust and understanding of the system (Stumpf et al., 2009). In this context, our approach stands out as we provide a visualization of topic clusters to the annotators to facilitate their task.

3 Methodology

Our active learning pipeline consists of three stages, intent classification, error detection and topic modeling. The full pipeline is depicted in Figure 1 as a block diagram.

Intent Classification This is the production model that predicts the intent of the user utterance. Due to the scalability constraints in terms of latency and computing resources, this model must have low inference time and run on CPUs. Without loss of generality, in this work we employ the Universal Sentence Encoder (USE) (Cer et al., 2018) as embedder, followed by linear Support Vector Classification (SVC). During live conversations, both the user utterance and the predicted intent are stored and passed to the next stage of the pipeline for offline error detection.

Error Detection We fine-tune two LLMs for intent classification with the same training set used to train the production classifier. Then, for each utterance collected in production, we predict their intent with the two LLMs and compare these results with the intent predicted by the production model. If there is disagreement between the three intents we mark the utterance as problematic. The LLMs used are DistilBERT (Sanh et al., 2019) and DeBERTa-v3-base (He et al., 2021b,a) since they differ significantly in size and pretraining objectives, which diversifies the predictions of hard-to-classify utterances.

Clustering and Topic Modeling We divide the set of utterances marked as problematic in the previous stage by the intent given in production. Then for each intent, we perform clustering and topic modeling following a similar approach to BERTopic (Grootendorst, 2022) but with USE embeddings. We use UMAP (McInnes et al., 2018) for dimensionality reduction, HDBSCAN (McInnes et al., 2017) for clustering and c-TF-IDF for topic modeling i.e. for generating topic keywords that help the annotators to categorize the error type within the cluster. We perform a random search

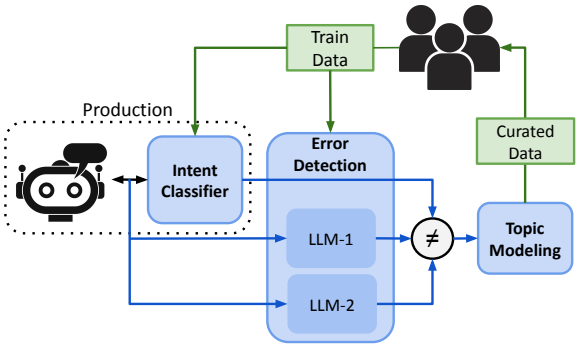

Figure 1: Active learning pipeline: the utterances received by the virtual assistant at production are passed through an intent classifier, error detection and topic modeling to create a curated dataset that is labeled by human experts and integrated in the training data.

to select hyperparameters and pick the combination that minimizes the amount of data points labeled as noise. Formally, we want to minimize the proportion of data clustered with confidence score smaller than 0.05.

For each intent, the topic modeling stage returns a set of clusters of problematic utterances with three topic words describing the cluster. This is the final output of our pipeline which is then given to the human experts for analysis and annotation. This way, the human experts receive a curated and ordered set of potentially critical utterances that can be quickly labeled and integrated in the training set of the production model.

4 Experiments and Results

Here, we conduct a quantitative and a qualitative evaluation. In our quantitative evaluation we assess to what extent our pipeline reduces the amount of labeled data needed to reach certain performance; and in the qualitative evaluation we analyze discovered error patterns. We run our experiments on two public datasets: ATIS (Hemphill et al., 1990) and SNIPS (Coucke et al., 2018); and an internal dataset (AUTO) consisting of real-world data from the automotive domain. ATIS is a dataset of queries about flight information of 4,978 training samples and 893 test samples[1]. SNIPS is a dataset of interactions between users and virtual assistants like Siri or Alexa. We use the version with 26,000 utterances and we set as label the joint fields "intent" and "scenario", which results in 64 classes.

[1]We use: https://github.com/microsoft/CNTK

4.1 Data Annotation

We simulate a real-world scenario where a production model f_{prod} classifies the intent of a large number of samples. This prediction is combined with that of two other models f_{err1} and f_{err2} to perform offline error detection. As explained in Section 3, f_{prod} consists of USE for embedding followed by linear SVC, while f_{err1} and f_{err2} are DistilBERT and DeBERTa-base respectively.

To simulate our production setting for a given dataset, we perform a 10-90 split of the training data, where we use the 10% split to train the initial model. This corresponds to the first model deployed in production, trained with a small amount of initially available data. The remaining 90% of the train data simulates the data progressively acquired in production. We follow an iterative process with each iteration corresponding to an annotation campaign where human experts annotate a set of production samples. These samples are incorporated to the training data of f_{prod}, which is then re-trained with the expanded training set. At each iteration i, we denote the training data as D_i^{train} and the rest as D_i^{rest}. Furthermore, we use the held-out test set D^{test} to assess the performance of the intent classification model at each iteration.

In detail, each iteration i starts with training the intent classification model f_{prod} and fine-tuning the error detection models f_{err1} and f_{err2} with D_i^{train}. At this point, to keep track of the evolution of the performance of the model, we evaluate f_{prod} on D^{test} by computing the macro-averaged F1 score. Then, we predict with the three models the intent of 15%[2] of D_i^{rest}. Those samples for which the three models do not agree on the prediction are added with their ground-truth labels to D_{i+1}^{train} and removed from D_{i+1}^{rest}, this simulates the annotation by human experts. The process is repeated until no new data is added to D_{i+1}^{train}.

In Table 2, we report for each dataset the macro F1 score obtained by f_{prod} when training with 100% of the data as well as the percentage of data needed to reach the same performance (within a ±0.005 error) with our active learning pipeline (AL). We also report the maximum F1 attained with our pipeline and the percentage of train data needed to reach it. The results shown are the

[2]15% is an arbitrary amount to simulate incoming data. Proportions like 5% or 10% would serve the same purpose.

Dataset	Topic	Examples	Ground Truth	Predicted Intent
ATIS	flights, flight, Denver	*How much is a flight from Washington to Montreal*	flight	airfare
	flights, flight, Denver	*What is the airfare for flights from Denver to Pittsburgh on Delta airline*	flight	airfare
	flights, flight, Denver	*List airlines that fly from Seattle to Salt Lake City*	flight	airline
	flights, flight, Denver	*Please show me airlines with flights from Denver to Boston with stop in Philadelphia*	flight	airline
SNIPS	events, calendar, today	*When is my next dentist appointment*	query_event_calendar	delete_event_calendar
	events, calendar, today	*Show up the events for me today*	query_event_calendar	delete_event_calendar
	events, calendar, today	*Tell me what is on my calendar for tomorrow*	query_event_calendar	delete_event_calendar
	meeting, hour, remind	*Remind me about the meeting tomorrow at six*	set_reminder	notification_calendar
	meeting, hour, remind	*Schedule a reminder one hour before the meeting*	set_reminder	notification_calendar

Table 1: Examples of error patterns discovered per dataset by our pipeline.

Dataset	100% Data	% Match AL	Max AL	% Max AL
ATIS	0.699	25.6	0.725	26.6
SNIPS	0.745	54.8	0.745	54.8
AUTO	0.784	36.0	0.795	35.7

Table 2: Results of the data annotation experiments; performance numbers are macro F1 scores. *% Match AL* is the amount of data labeled by the active learning (AL) pipeline that matches the *100% Data* score; *Max AL* is the maximum performance reached with AL and *% Match AL* is the amount of data to get that score.

mean across five different splits of the data.

For the three datasets, the amount of data needed to match the performance of the full training set with our pipeline (AL) is much smaller. In particular, for ATIS we need only 25.6% of the data, for SNIPS 54.8% and for AUTO 36.0%. Furthermore, for ATIS and AUTO we outperform the model trained with the full train set with only 26.6% and 35.7% of the data respectively. These results demonstrate the large savings in terms of data annotation that can be obtained with our pipeline, which in turn can represent a major reduction in costs for an organization.

4.2 Error Analysis

Next, we conduct a qualitative analysis of the error patterns discovered by our pipeline, similar to the analysis that would be performed by human experts during error exploration. We report results for the two public datasets, ATIS and SNIPS. For each dataset, we simulate an imperfect production classifier by training f_{prod} on 50% of the data. Then, we run intent classification, error detection and topic modelling on the remaining 50% of the data, as well as, on the test set. We manually analyze the clusters produced to understand where the model is failing and in Table 1 we report some patterns discovered in this way.

For ATIS, some utterances that should be classified as "flight" are mis-classified as "airfare" or "airiline", while for SNIPS, we see that instead of querying the calendar, the model is misunderstanding to delete events, and instead of setting reminders it is adding notifications. We argue that certain intent mis-classifications, such as the ones shown here, can be critical for the operation of a virtual assistant and should be detected as early as possible.

The analysis shown in this section requires little technical knowledge for the human experts, since they only need to look at the generated clusters and assess which ones represent a major risk. This can greatly speed up the error analysis process, helping in the early detection of critical errors and in reducing the amount of time that the annotators need to spend looking at the data.

5 Conclusion

In this work we have presented an active learning pipeline for conversational agents which consists of intent classification, unsupervised error detection and topic modeling. In the experiments, we show that our approach helps in prioritizing data for annotation: in our real-world dataset (AUTO) we reach the same performance with 36% of the data when selected by our pipeline as with 100% without prioritization. Therefore, this method can provide major savings for organizations with limited annotation capabilities. Furthermore, we argue that our approach helps to discover intent classification errors that may be critical for the correct operation of the dialogue agent and which, if not detected on time, could jeopardize the viability of the system. In future work, we plan to extend our proposed pipeline to support also named entity recognition.

References

Ron Artstein. 2017. Inter-annotator agreement. *Handbook of linguistic annotation*, pages 297–313.

Josh Attenberg and Foster Provost. 2010. Why label when you can search? alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 423–432.

Joshua Attenberg, Panos Ipeirotis, and Foster Provost. 2015. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *Journal of Data and Information Quality (JDIQ)*, 6(1):1–17.

Akansha Bhardwaj, Jie Yang, and Philippe Cudré-Mauroux. 2020. A human-ai loop approach for joint keyword discovery and expectation estimation in micropost event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2451–2458.

Akansha Bhardwaj, Jie Yang, and Philippe Cudré-Mauroux. 2022. Human-in-the-loop rule discovery for micropost event detection. *IEEE Transactions on Knowledge and Data Engineering*.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Nan-Chen Chen, Jina Suh, Johan Verwey, Gonzalo Ramos, Steven Drucker, and Patrice Simard. 2018. Anchorviz: Facilitating classifier error discovery through interactive semantic data exploration. In *23rd International Conference on Intelligent User Interfaces*, pages 269–280.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Rinat Khaziev, Usman Shahid, Tobias Röding, Rakesh Chada, Emir Kapanci, and Pradeep Natarajan. 2022. Fpi: Failure point isolation in large-scale conversational assistants. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 141–148.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. 2017. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

David D Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29).

Besmira Nushi, Ece Kamar, Eric Horvitz, and Donald Kossmann. 2017. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International journal of human-computer studies*, 67(8):639–662.

Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. 2001. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292.

Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. 2011. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168.

Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. 2018. Leveraging crowdsourcing data for deep active learning an application: Learning intents in alexa. In *Proceedings of the 2018 World Wide Web Conference*, pages 23–32.