

ADAPTIVE ENDPOINTING WITH DEEP CONTEXTUAL MULTI-ARMED BANDITS

Do June Min^{1*}, Andreas Stolcke², Anirudh Raju², Colin Vaz²,
Di He², Venkatesh Ravichandran² and Viet Anh Trinh²

¹University of Michigan ²Amazon Alexa AI, USA
dojmin@umich.edu {stolcke, ranirudh, vazcoli, deehe, veravic, trinhvie}@amazon.com

ABSTRACT

Current endpointing (EP) solutions learn in a supervised framework, which does not allow the model to incorporate feedback and improve in an online setting. Also, it is a common practice to utilize costly grid-search to find the best configuration for an endpointing model. In this paper, we aim to provide a solution for adaptive endpointing by proposing an efficient method for choosing an optimal endpointing configuration given utterance-level audio features in an online setting, while avoiding hyperparameter grid-search. Our method does not require ground truth labels, and only uses online learning from reward signals without requiring annotated labels. Specifically, we propose a deep contextual multi-armed bandit-based approach, which combines the representational power of neural networks with the action exploration behavior of Thompson modeling algorithms. We compare our approach to several baselines, and show that our deep bandit models also succeed in reducing early cutoff errors while maintaining low latency.

Index Terms— endpointing, multi-armed bandits, automatic speech recognition, turn taking, dialog modeling.

1. INTRODUCTION

In modern spoken language AI assistants and dialog systems, endpointing is a key step in the system pipeline, determining when a speaker has finished an utterance [1, 2, 3, 4]. Similar to turn-taking in human-human conversations, smooth endpointing that avoids early cutoffs of speaker utterances or excessive latency before an agent response is key to efficient conversational interaction [5]. For instance, speech disfluencies in the form of pauses can lead to poor endpointing, and require attention to prosodic properties to avoid mistaking them for utterance-final pauses [2]. Regardless of the modeling used, endpointing hyperparameters need to be carefully calibrated, e.g., to find a good balance between early cutoffs and latency [6, 7, 8].

To simplify the problem for learning purposes, in this paper we investigate learning the choice between just two endpointing configurations, “standard” and “relaxed”, using features that are extracted for each speaker or utterance (i.e., a sequence of utterances). Whereas the “standard” configuration leads to endpointing behavior suitable for most speakers, the “relaxed” configuration is suited for utterances with slow speaking rate and more mid-utterance pausing. Thus, the task of endpointing adaptation is formulated as finding the better configuration for each utterance. Although considerable work has been done on different endpointing models and algorithms, there have been few studies on how endpointing hyperparameters can be optimized at a personal or contextual level [9, 10, 11]. Furthermore, adaptive decision making in acoustic modeling has been

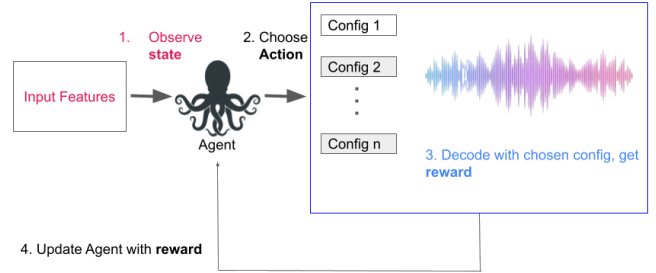


Fig. 1. An overview of our adaptive endpointing with deep contextual multi-armed bandit (CMAB).

studied mostly in the context of ASR [12, 13], rather than endpointing.

Specifically, our goal is to address the following questions:

Which features are most useful? As inputs to the adaptive endpointing model, different types of features can be considered. Features that can be extracted directly from the target audio are good candidates. Although it is not realistic to assume the full target audio would be available in time for adaptive endpointing, it is reasonable to assume that features derived from the initial parts of an utterance can be fed to an online model. We compare these features and how they impact the performance of our models.

How much information do we need? Since an endpointing agent needs to choose a configuration before the entire input has been consumed, the task is to predict in advance whether an early cutoff is likely, rather than detecting an event that has already happened. Clearly the more of the input the agent sees, the more accurate the predictions will be. In simulations we investigate the effects of varying amounts of prior data for making a decision about the endpointing configuration.

Can an online contextual bandit model be used in place of an offline-trained model? Finally, we note that a supervised learning framework is not suitable for online learning, and investigate whether an online model may be used instead. Specifically, we adopt the contextual multi-armed bandit (CMAB) framework so that models can learn from reward signals based on latency and cutoff results, instead of ground-truth annotations.

In summary, we find that (1) target audio and partial ASR hypotheses based on the starts of utterances are most important; (2) the more target audio data, the better the performance up to a point—with only about an initial 20% of the data, an agent can reduce early cutoff without degrading latency—; and (3) online models such as deep CMAB are applicable to the endpointing task, reducing cutoffs while maintaining latency performance.

*This work was done during the author’s Amazon internship.

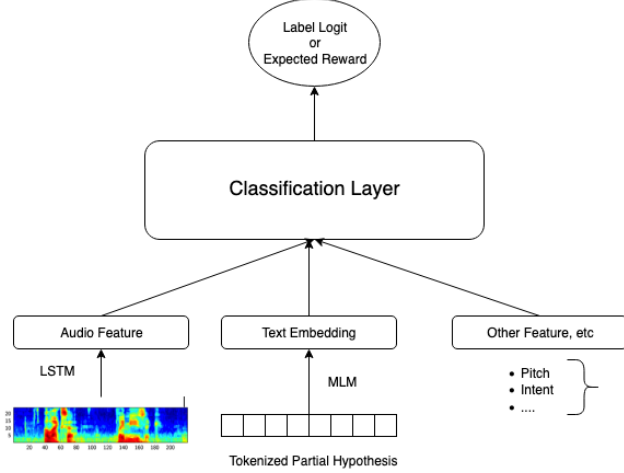


Fig. 2. Shared model architecture for the supervised classifier and deep contextual bandit models. For the bandit models, the classification layer is replaced by a reward predictor for each action.

2. TASK & METHODOLOGY

2.1. Task: Adaptive endpointing

Our task is to predict the optimal endpointing configuration for the speaker. While there could be an unlimited number of configurations for each hyperparameter set, we limit our attention to this binary setting, where the focus is to reliably predict when the target utterance is better endpointed with the “relaxed” configuration, as opposed to the default or “standard” configuration.

2.2. Dataset

For our study, we use de-identified data sampled from a voice-enabled assistant. Using this collection of utterances, we then annotate each utterance with ground truth information using the following logic:

- If an utterance is cut off early with the “standard” configuration, label the utterance as Class 1, meaning that the optimal configuration for the utterance is “relaxed” (positive class).
- Conversely, if an utterance is not cut off early, then the utterance is labeled “standard”, or Class 0.

We split our collection of about 610 hours into training, development, and test splits with a ratio of 8:1:1, each with about 2.5% positive labels (early cut-off with “standard” configuration). Overall, only about 0.02% of the utterances are cut off early in both of the configurations. The audio data is in English.

2.3. System Architecture & Features

Our model architecture is an LSTM-based model proposed by Maas et al, which uses acoustic features and is pretrained to predict both end-of-utterance and voice activity [9]. We make necessary modifications by adding a last-frame pooling step, and add an MLM-based text encoder to embed partial text hypotheses. The audio and hypothesis features are concatenated with additional (“Other”) features and fed to the classification layer for prediction.

Table 1. Input Features

Name	Description
Audio	extracted audio features
Hypothesis	best-1 hypothesis from decoding
Pause Duration	time between wakeword and intent
Wakeword Duration	duration of wakeword
Pitch Features	paralinguistic features for intonation
Intent Domain	domain of the intentful utterance

The full set of features we experimented with is listed in Table 1. Audio and hypothesis features represent the acoustic and semantic content of the utterance, respectively. Language model features model the syntactic and semantic completeness of utterances and have been shown to boost the performance of endpointing models [1]. While transcription hypotheses are distinct from language model posteriors typically used in such studies, we take partial or complete transcriptions of the target audio as a proxy for language model predictions. In addition, we also include some hand-crafted features that are relevant to endpointing based on prior work. Wakeword and pause duration could be indicative of initial speaking rate or hesitation by the speaker, while past research shows that prosodic and paralinguistic features, such as pitch, are important for endpointing [1, 14, 15, 16, 17, 18]. Also, we assume oracle access to the intent domain, which refers to the category of the user’s command. We included this feature based on our intuition that certain intents or commands are more likely to induce slower or more disfluent speech production, such as web search or question answering.

2.4. Deep contextual multi-armed bandit (Deep CMAB) algorithm

A key disadvantage of the supervised approach is that it cannot be trained in an online manner, since training requires knowing the decoding result with the “standard” configuration, regardless of which decision it made during the prediction step, making it necessary to prepare the training set in advance. On the other hand, online learning frameworks such as CMAB only require that the agent receive a reward signal, for the chosen action [19].

Online bandit models such as Linear Thompson sampling have been successfully applied in online prediction settings such as recommendation systems [20], and recent studies show that moving from offline models trained on custom datasets to online models trained using implicit signal can provide significant improvement in performance and cost [21].

Thus, we adopt the CMAB approach for our online agents. Specifically, we adopt the recent deep implementation of the CMAB, instead of popular linear algorithms such as Linear Thompson sampling [22]. We find that deep bandits are better suited to our task both for their representational power and for handling audio and their batched training capability. While there are different algorithms for deep bandits, such as proposed in [23, 24], we adopt the relatively simple framework proposed in [25]. One difference between [25] and our approach is that instead of updating the model periodically, we update our model after seeing each batch of examples. Specifically, we adopt concrete dropout in place of conventional dropout weights for neural network model training. Gal et al. show that concrete dropout allows the model to calibrate the amount of exploration naturally, as training progresses [26].

The pseudocode for our algorithm is given below as Algorithm 1. The neural network model uses the architecture shown in

Algorithm 1: Deep Contextual Bandit Pseudocode

```
Data: Set of utterances  $S$ , Neural Network  $f$ 
for  $utterances \in S$  do
  for  $a_i \in \text{Actions}$  do
     $rewards_i \leftarrow f(s, a_i)$ 
  end
   $chosen \leftarrow \text{choose\_action}(rewards)$ 
   $real\_reward \leftarrow \text{decode}(s, chosen)$ 
   $f \leftarrow \text{update}(f, real\_reward, rewards)$ 
  if  $terminate$  then
    break
  end
end
```

Figure 2, while for **choose_action**, we take the greedy action with argmax, and **update** the model through stochastic gradient reward. In the implementation, our model predicts rewards for both actions simultaneously, rather than predicting a reward given an action. For the bandit model, the reward signal is computed as a linear combination of latency (in ms) and cutoff (indicator variable), which is then used in the reward prediction loss calculation (mean squared error loss). The mixing weights are hyperparameters that we find using experiments on held-out development data. Intuitively, the bandit model tries to predict the expected reward of each configuration, and chooses a configuration based on the reward, while the supervised classifier directly outputs the predicted optimal configuration.

3. EXPERIMENTS

3.1. Experiment setup

Beyond accuracy measures, we also evaluate the performance of our models using the following metrics: (1) Early endpointing rate (Early EP rate), which measures the fraction of times the endpointer triggers before the end of the last utterance is reached. (2) Trimmed mean 95 (TM95) of latency (ms), average of the lower 95th percentile of the data (3) Double-trimmed mean 95 (DTM95:99) of latency (ms), average of the interval (inclusive) between the 95th and 99th percentiles of the data.

To contextualize and better compare the performance of our models, we also measure the results of several baselines. **Standard Only**, the default baseline model, always chooses the “standard” configuration, while **Relaxed Only** always chooses the “relaxed” configuration. Lastly, we also consider the **Oracle Model**, which always outputs the optimal configuration choice, giving an upper bound of achievable performance.

3.2. Which features are most useful?

To study which features are important for adaptive endpointing, we conduct utterance-wise endpointing experiments in an idealized setting. That is, we assumed that the adaptive endpointing model will have the full length of the target audio and the features derived from the full audio as inputs. In Table 2, we note that target audio and hypothesis features achieve largest cutoff reduction, followed by intent domain, which incurs significant latency degradation.

We note that the top-performing features (target audio, target hypothesis, and intent domain) require processing of the target audio utterance. On the other hand, features that can be obtained without

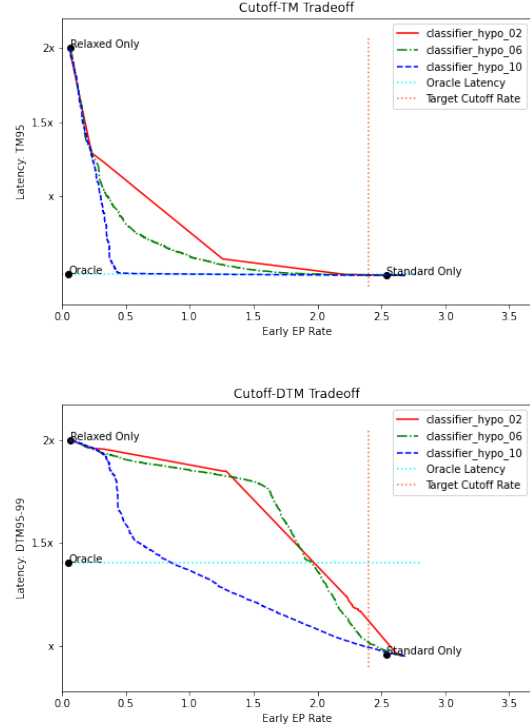


Fig. 3. Comparing hypothesis features with varying portions on cutoff-latency tradeoff curves.

the intent-carrying portion of an utterance (wakeword duration, pitch features) perform poorly, showing that it is difficult to reliably predict the overall utterance pattern just from the paralinguistic features derived from the initial parts of the utterance.

Conclusion: Hypothesis and audio features are most informative. In an idealized setting where the whole target utterance is used for prediction, hypothesis and audio features lead to 80% and 20% relative cutoff rate reduction, respectively, with no TM95 degradation and Oracle-level DTM95:99 latency.

3.3. How much information do we need?

In practice, consuming the whole target audio is unrealistic. Thus, we relax this assumption by assuming our models now see data corresponding to the first $X\%$ of the utterance. To derive the input features for this setup, we first process the full target utterance to derive corresponding audio and hypothesis features. Then, we take the first $X\%$ of the resulting features and feed them to our model. We conducted experiments to compare bandit models trained with all types of features, but we chose to include only the two features with the best performance, for brevity and legibility.

Figure 3 shows the latency vs. early EP trade-off curves plotted for supervised classifiers with target hypothesis features with varying amount (20%, 60%, 100%) of tokens. First, we observe that the cutoff vs. DTM95:99 curve has a “worse” trade-off curve since more latency degradation is required to achieve the same amount of early cutoff reduction. This is because by definition of DTM95:99 is more sensitive to changes in the tail of the latency distribution. However, we note that even the 20% model achieves no TM95 latency degradation and only $\sim 20\%$ DTM95:99 latency degradation, as indicated

Table 2. Endpointing metrics obtained with different features in an idealized setting where the features are computed from the whole target audio. Relative results (indicated by \pm) use the Standard Only as a baseline. TM95 refers to a trimmed mean (lower 95th percentile). DTM95:99 refers to a doubly-trimmed mean (95th to 99th percentile).

Metrics / Model	Standard Only	Relaxed Only	Oracle Result	Target Audio	Target Hypothesis	Intent Domain	Wakeword Duration	Pitch Features	Pause duration
Accuracy (%)	97.46	2.75	100	87.48	97.68	21.11	95.23	94.40	67.43
Precision (%)	NA	2.75	100	17.19	53.50	2.29	0.40	3.66	2.46
Recall (%)	NA	100	100	91.57	84.83	68.41	0.29	4.07	28.67
F1 score	NA	5.36	100	28.95	65.61	4.44	0.34	3.86	4.53
Early EP rate	-	0.07	0.05	0.29	0.43	0.89	2.77	2.65	1.94
Early EP rate change	-	-97.24%	-98.03%	-88.58%	-83.07%	-64.96%	+9.06%	+4.33%	-23.62%
Latency (TM95)	-	+323.42%	+1.65%	+29.75%	+3.03%	+257.58%	+0.28%	+3.03%	+95.59%
Latency (DTM95:99)	-	+108.92%	+46.51%	+95.66%	+67.47%	+106.99%	+27.59%	+64.34%	+100.24%

by the intersection between the blue curves and the dotted red line (target cutoff rate).

Moreover, we confirm that using a larger portion of the target features improves the performance. In Figure 3, the 100% model achieves significantly better trade-offs for both TM95 and DTM95:99. This is a confirmation of the intuition that the latter parts of the audio provide more information about which endpointing configuration is optimal for the audio.

Conclusion: By using only the initial 20% of the data, we can reduce Early-EP rate by 5%, with no degradation in TM95 and 20% relative DTM95:99 degradation over the baseline. However, we also find that the latter parts of the contain valuable information about optimal endpointing configurations.

3.4. Can an online contextual bandit model be used in place of an offline-trained model?

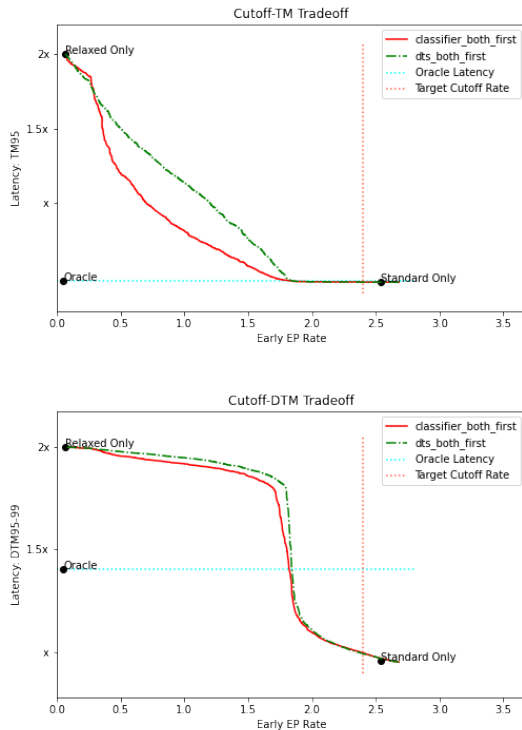


Fig. 4. Comparing supervised and bandit models on first segment input features.

Since the offline setup required by the supervised model is not ideal, as discussed in Section 2.4, we investigate if a deep CMAB model can also meet the objective of reducing early cutoff without degrading latency, by comparing the supervised and bandit models. Furthermore, we compare these predictive models in a more realistic setting where the total length of the target audio is not known beforehand. (In previous experiments, we either assumed an idealized setting, with access to the full target utterance, or knowing the utterance length in advance.) Hence, we extracted the initial segment of an utterance containing the wake word, and use the audio and hypothesis corresponding only to that segment. On average, the time fraction of that initial segment is $\sim 30\%$ of the full utterance. We consider this setup a simulation of when a speculative listener is first activated, and retrieves a partial utterance and decoding result (hypothesis) to the endpointing model.

Figure 4 shows the trade-off curves for the supervised and bandit models. While the bandit model (dts_both_first) achieves slightly worse trade-offs for both TM95 and DTM95:99, both models achieve target cutoff reductions with little (DTM95:99) to no (TM95) sacrifice for latency measures. We note that both supervised and bandit models achieve a significant ($2.5 \rightarrow 1.8$, $\sim 30\%$) Early-EP rate reduction without any TM95 latency degradation, while a small degradation in DTM95:99 latency is observed.

Conclusion: Deep contextual bandits can reduce cutoff rate by 5% without TM95 latency degradation, and $\sim 20\%$ DTM95:99 degradation. The more target data is available, the more the gap between supervised classifier and bandit model narrows.

4. CONCLUSIONS

We have proposed adaptive endpointing as a framework for dynamically choosing an optimal endpointing configuration, based on features derived from each input utterance. By implementing a static supervised classifier for endpointing configuration, we show that utterance-level selection of the locally best endpointing configuration leads to reduction in early cutoff rate, while keeping latency degradation small. We also show that an online model can be trained without having access to ground truth data. For this purpose, a deep contextual multi-armed bandit (CMAB) model combines the efficiency of Bayesian exploration with the representational power of neural networks, does not require ground truth annotation, and can be adapted to utilize a variety of reward signals that may be available in an online deployment setting. We find that audio and text features derived from the target utterance are most important for endpointing, and online-trained deep CMAB models can be used in place of impractical offline supervised classifiers, while still reducing early cutoff without latency degradation.

5. REFERENCES

- [1] L. Ferrer, E. Shriberg, and A. Stolcke, "A prosody-based approach to end-of-utterance detection that does not require speech recognition," in *Proc. IEEE ICASSP*, 2003, vol. 1, pp. I–I.
- [2] Harish Arsikere, Elizabeth Shriberg, and Umut Ozertem, "Computationally-efficient endpointing features for natural spoken interaction with personal-assistant systems," in *Proc. IEEE ICASSP*, 2014, pp. 3241–3245.
- [3] Siyan Li, Ashwin Paranjape, and Christopher D. Manning, "When can I speak? Predicting initiation points for spoken dialogue agents," arXiv:2208.03812, 2022.
- [4] Shuo-yiin Chang, Bo Li, Tara N. Sainath, Chao Zhang, Trevor Strohman, Qiao Liang, and Yanzhang He, "Turn-taking prediction for natural conversational speech," 2022.
- [5] David Schlangen, "From reaction to prediction: Experiments with computational models of turn-taking," in *Proc. Interspeech*, Pittsburgh, 2006, pp. 2010–2013.
- [6] Yingzhu Zhao, Chongjia Ni, Cheung-Chi Leung, Shafiq Joty, Eng Siong Chng, and Bin Ma, "Preventing early endpointing for online automatic speech recognition," in *Proc. IEEE ICASSP*, 2021, pp. 6813–6817.
- [7] Liang Lu, Jinyu Li, and Yifan Gong, "Endpoint detection for streaming end-to-end multi-talker ASR," in *Proc. IEEE ICASSP*, 2022, pp. 7312–7316.
- [8] W. Ronny Huang, Shuo yiin Chang, David Rybach, Rohit Prabhavalkar, Tara N. Sainath, Cyril Allauzen, Cal Peyser, and Zhiyun Lu, "E2E Segmenter: Joint segmenting and decoding for long-form ASR," in *Proc. Interspeech*, 2022.
- [9] Roland Maas, Ariya Rastrow, Chengyuan Ma, Guitang Lan, Kyle Goehner, Gautam Tiwari, Shaun Joseph, and Björn Hoffmeister, "Combining acoustic embeddings and decoding features for end-of-utterance detection in real-time far-field speech recognition systems," in *Proc. IEEE ICASSP*, 2018, pp. 5544–5548.
- [10] Aditya Jayasimha and Periyasamy Paramasivam, "Personalizing speech start point and end point detection in ASR systems from speaker embeddings," *Proc. IEEE Spoken Language Technology Workshop*, pp. 771–777, 2021.
- [11] Shaojin Ding, Rajeev Vijay Rikhye, Qiao Liang, Yanzhang He, Quan Wang, Arun Narayanan, Tom O'Malley, and Ian McGraw, "Personal VAD 2.0: Optimizing personal voice activity detection for on-device speech recognition," in *Proc. Interspeech*, 2022.
- [12] Tsendsuren Munkhdalai, Khe Chai Sim, Angad Chandorkar, Fan Gao, Mason Chua, Trevor Strohman, and Françoise Beaufays, "Fast contextual adaptation with neural associative memory for on-device personalized speech recognition," in *Proc. IEEE ICASSP*, 2022, pp. 6632–6636.
- [13] Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant Strimel, Athanasios Mouchtaris, and Siegfried Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *Proc. IEEE ICASSP*, 2022.
- [14] Roland Maas, Ariya Rastrow, Kyle Goehner, Gautam Tiwari, Shaun Joseph, and Björn Hoffmeister, "Domain-specific utterance end-point detection for speech recognition," in *Proc. Interspeech*, 08 2017, pp. 1943–1947.
- [15] Yuichi Ishimoto, Takehiro Teraoka, and Mika Enomoto, "End-of-utterance prediction by prosodic features and phrase-dependency structure in spontaneous Japanese speech," in *Proc. Interspeech*, 2017.
- [16] Chaoran Liu, Carlos Toshinori Ishi, and Hiroshi Ishiguro, "Turn-taking estimation model based on joint embedding of lexical and prosodic contents," in *Proc. Interspeech*, 2017.
- [17] Samuel Thomas, Sri Harish Reddy Mallidi, Thomas Janu, Hynek Hermansky, Nima Mesgarani, Xinhui Zhou, Shihab A. Shamma, Tim Ng, Bing Zhang, Long Nguyen, and Spyridon Matsoukas, "Acoustic and data-driven features for robust speech activity detection," in *Proc. Interspeech*, 2012.
- [18] Angelika Maier, J. Hough, and David Schlangen, "Towards deep end-of-turn prediction for situated spoken dialogue systems," in *Proc. Interspeech*, 2017.
- [19] Shipra Agrawal and Navin Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *Proc. ICML*, 2013.
- [20] Fabian Moerchen, Patrick Ernst, and Giovanni Zappella, "Personalizing natural language understanding using multi-armed bandits and implicit feedback," in *Proc. 29th ACM International Conference on Information and Knowledge Management*, New York, NY, USA, 2020, CIKM '20, p. 2661–2668.
- [21] Ge Yu, Chengwei Su, and Emre Barut, "Introducing deep reinforcement learning to NLU ranking tasks," in *Proc. IEEE ICASSP*, 2021.
- [22] Daniel Russo, Benjamin Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen, "A tutorial on Thompson sampling," *Foundations and Trends in Machine Learning*, vol. 11, no. 1, pp. 1–96, 2018.
- [23] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu, "Neural Thompson sampling," in *Proc. International Conference on Learning Representations*, 2021.
- [24] Carlos Riquelme, George Tucker, and Jasper Snoek, "Deep Bayesian bandits showdown: An empirical comparison of bayesian deep networks for Thompson sampling," in *Proc. 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018, OpenReview.net.
- [25] Mark Collier and Hector Urdiales Llorens, "Deep contextual multi-armed bandits," arXiv:1807.09809, 2018.
- [26] Yarin Gal, Jiri Hron, and Alex Kendall, "Concrete dropout," in *Proc. 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, NIPS'17, p. 3584–3593, Curran Associates Inc.