Optimizing NLU Reranking Using Entity Resolution Signals in Multi-domain Dialog Systems

Tong Wang[*], Jiangning Chen[*], Mohsen Malmir, Shuyan Dong,
Xin He, Han Wang, Chengwei Su, Yue Liu, Yang Liu
Amazon Alexa
{tonwng, cjiangni, malmim, shuyand, xih, wnghn, chengwes, lyu, yangliud}@amazon.com

Abstract

In dialog systems, the Natural Language Understanding (NLU) component typically makes the interpretation decision (including domain, intent and slots) for an utterance before the mentioned entities are resolved. This may result in intent classification and slot tagging errors. In this work, we propose to leverage Entity Resolution (ER) features in NLU reranking and introduce a novel loss term based on ER signals to better learn model weights in the reranking framework. In addition, for a multi-domain dialog scenario, we propose a score distribution matching method to ensure scores generated by the NLU reranking models for different domains are properly calibrated. In offline experiments, we demonstrate our proposed approach significantly outperforms the baseline model on both single-domain and cross-domain evaluations.

1 Introduction

In spoken dialog systems, natural language understanding (NLU) typically includes domain classification (DC), intent classification (IC), and named entity recognition (NER) models. After NER extracts entity mentions, an Entity Resolution (ER) component is used to resolve the ambiguous entities. For example, NLU interprets an utterance to Alexa (or Siri) "play hello by adele" as in the 'Music' domain, 'play music' intent, and labels "hello" as a song name, "adele" as an artist name. ER queries are then formulated based on such a hypothesis to retrieve entities in music catalogs. Often times NLU can generate a list of hypotheses for DC, IC, and NER, and then a reranking model uses various confidence scores to rerank these candidates (Su et al., 2018).

Since ER is performed after NLU models, the current NLU interpretation of the utterance is limited to the raw text rather than its underlying entities. Even in NLU reranking (Su et al., 2018), only DC, IC, and NER confidence scores were used, and as a result, the top hypothesis picked by NLU reranking might not be the best interpretation of the utterance. For example, in the absence of entity information, "the beatles" in the utterance "play with the beatles" is interpreted as an artist name. If the reranker could search the ER catalog, it would promote the hypothesis that has "with the beatles" as an album name. Such NLU errors may propagate to ER and downstream components and potentially lead to end-customer friction.

In this work, we thus propose to incorporate ER features in the NLU reranking model, called NLU-ER reranking. For a domain, we use its corresponding catalogs to extract entity related features for NLU reranking for this domain. To enhance ER feature learning, we add a novel loss term when an NER hypothesis cannot be found in the catalog. One additional challenge arises in the multi-domain systems. In large-scale NLU systems, one design approach is to modularize the system as per the concept of domains (such as Music, Video, Smart Home), and each domain has its own NLU (DC, IC, NER) and reranking models that are trained independently. Under this scheme, each domain's NLU reranking plays an important role in both in-domain and cross-domain reranking, since it not only ranks hypotheses within a domain to promote the correct hypothesis, but also produces ranking scores that need to be comparable across all different domains. In (Su et al., 2018), the scores for the hypotheses from different domains are calibrated through training on the same utterance data with similar models . However, we may only use NLU-ER reranking for some domains (due to reasons such as lack of entity catalog, different production launch schedule, etc.), and the scores from such rerankers may no longer be comparable with other domains using the original reranker model. To mitigate this issue, we introduce a score distribution matching method to adjust the score distributions.

[*]The first two authors have equal contribution

We evaluate our NLU-ER reranking model on multiple data sets, including synthetic and real dialog data, and for both single domain and cross-domain setups. Our results show improved NLU performance compared to the baseline, and the improvement is contributed to our proposed ER features, loss term, and score matching method.

2 Related Work

Early reranking approaches in NLU systems use a single reranker for all the domains. Robichaud et al. (Robichaud et al., 2014) proposed a system for multi-domain hypothesis ranking (HR) that uses LambdaMART algorithm (Burges et al., 2007) to train a ranking system. The features in the ranking system include confidence scores for intents and slots, relevant database hits and contextual features that embed relationship to previous utterances. The authors showed improved accuracy in top domains using both non-contextual and contextual features. Crook et al. adapted a similar reranking scheme for multi-language hypothesis ranking (Crook et al., 2015). The set of features in the reranker include binary presence variables, for example presence of an intent, coverage of tagged entities and contextual features. They adapted the LambdaMART algorithm to train a Gradient Boosted Decision Trees model (Friedman, 2001) for cross language hypothesis ranking, and demonstrated comparable performance of the cross language reranker to the language-specific reranker. These models did not explicitly use ER signals for reranking. In addition, reranking is done across domains. Such single reranker approach is not practical in NLU systems with a large set of independent domains. In contrast, our approach emphasizes domain independence, allowing reranking to be performed for each domain independently. Furthermore, we rely on ER signal as a means to improve reranking.

To the best of our knowledge, the most related work to ours is Su et al. (Su et al., 2018), which proposed a re-ranking scheme to maximize the accuracy of the top hypothesis while maintaining the independence of different domains through implicit calibration. Each domain has its NLU reranker, and the scores for the hypotheses from reranking are compared across all the domains to pick the best hypothesis. The feature vector for each reranker is composed of intent, domain and slot tagging scores from the corresponding domain. Additionally, a cross entropy loss term is used to ensure calibration across domains. In a series of experiments, they demonstrated improvement of semantic understanding. Our work is an extension of that work as we utilize ER signals, in addition to the DC, IC, and NER scores, and introduce a new loss term to improve the reranking accuracy.

To resolve the score non-comparable problem in a multi-domain system, traditional calibration methods utilize Platt Scaling or Isotonic Regression to calibrate the prediction distribution into a uniform distribution (Zadrozny and Elkan, 2001, 2002; Platt et al., 1999; Niculescu-Mizil and Caruana, 2005; Wilks, 1990). However, this does not work in our scenario since the data in different domains are imbalanced, which causes domains with big traffic to have lower confidence scores. Instead of using probability calibration methods, we propose a solution based on power transformation to match the prediction score distribution back to the original score distribution, thus making the scores comparable even after ER information is added to NLU reranking.

3 Reranking Model

The baseline NLU reranking model is implemented as a linear function that predicts the ranking score from DC, IC, and NER confidence scores. We augment its feature vector using ER signals and introduce a novel loss term that penalizes the hypotheses that do not have a matched entity in the catalog. Similar to (Su et al., 2018), we tested using a neural network model for reranking, but observed no improvements, therefore we focus on the linear model.

3.1 ER Features in Reranking

The features used in the baseline NLU reranker include scores for DC (d), IC (i), NER (n) hypotheses, and ASR scores that are obtained from upstream components and used for all the domains. The additional ER features used in NLU-ER reranker are extracted and computed from the ER system, and can be designed differently for individual domains. For example, in this work, for the Music domain, ER features we use are aggregated from NER slot types such as: *SongName, ArtistName*, and the ER features are defined as:

ER success e_{s_i}: if a hypothesis contains a slot s_i that is successfully matched by any of the ER catalogs, this feature is set to 1, otherwise 0. ER success feature serves as a positive signal to pro-

mote the corresponding hypothesis score.

ER no match m_{s_i}: if a slot s_i in a hypothesis does not have any matched entities in the ER catalogs, this feature value is 1, otherwise 0. ER no match feature serves as a negative signal to penalize the hypothesis score. We find 'ER no match' is a stronger signal than 'ER success' because over 90% of the time, ER no match implies the corresponding hypothesis does not agree with the ground truth.

Similarity feature l_{s_i}: this feature is nonzero only if the ER success feature e_{s_i} is 1. In each catalog, a lexical or semantic similarity score between the slot value and every resolved entity is computed, and the maximum score among them is selected as the feature value. This indicates the confidence of the ER success signal.

Not in Gazetteer: this feature is set to 1 when ER features are not in the gazetteer (neither ER success nor no match), otherwise 0. We will discuss the gazetteer in the next section.

3.2 ER Gazetteer Selection

Since NLU and reranking happen before ER, in runtime retrieving ER features from large catalogs for NLU reranking is not trivial. Therefore we propose to cache the ER signals offline and make it accessible in NLU reranking in the form of a gazetteer. To make the best use of the allocated amount of runtime memory, we design a gazetteer selection algorithm to include the most relevant and effective ER features in the gazetteer.

We define Frequent Utterance Database (FUD) as the live traffic data where the same utterance has been spoken by more than 10 unique customers. To formalize the selection procedure, we define outperforming and underperforming utterances by friction (e.g., request cannot be handled) rate fr and 30s playback queue (playback $\geq$ 30s) rate qr. For all FUD utterances in a given period, an utterance u is defined as outperforming if $fr(u) \leq \mu_{fr} - \lambda_1 * \sigma_{fr}$ and $qr(u) \geq \mu_{qr} + \lambda_2 * \sigma_{qr}$, where μ and σ are the mean and standard deviation, λ_1 and λ_2 are hyperparameters. Underperforming utterances are defined likewise.

The detailed gazetteer selection algorithm is described in Algorithm 1. u_{h_1}, ..., u_{h_n} denote n-best NLU hypotheses of the utterance u. The idea is to encourage the successful hypotheses and avoid the friction hypotheses based on the historical data. For instance, if u is an underperforming utterance and u_{h_1} is *ER_NO_MATCH*, we want to penalize u_{h_1} to down-rank it, and promote other hypotheses u_{h_i} ($i \neq 1$) that receive the *ER_SUCCESS* signal. For the utterance hypotheses that are not selected in the gazetteer, we will use the Not_in_gazetteer (NG) feature.

Algorithm 1: Gazetteer Data Selection

```
Obtain outperforming and underperforming
  utterances from FUD;
for u ∈ outperforming utterances do
    if u_{h_1} is ER_SUCCESS then
        select ER features in u_{h_1} to the
          gazetteer;
    end
end
for u ∈ underperforming utterances do
    if u_{h_1} is ER_NO_MATCH then
        select ER features in u_{h_1} to the
          gazetteer;
    end
    if u_{h_i} is ER_SUCCESS, and h_i ≠ h_1
      then
        select ER features in u_{h_i} to the
          gazetteer;
    end
end
```

3.3 NLU-ER Reranker

For an utterance, the hypothesis score y is defined as the following:

$$y = W_G G + \sum_{s_i \in S} \mathbb{1}_{slot=s_i}(W_{s_i} ER_{s_i}) + \mathbb{1}_{NG} w_d \tag{1}$$

The first part in (1) is the baseline NLU reranker model:

$$y = W_G G \tag{2}$$

where $G = [g_1, g_2, \ldots, g_p]^T$ is the NLU general feature vector, $W_G = [w_1, w_2, \ldots, w_p]$ is the corresponding weight vector. The rest of the features are ER related. $\mathbb{1}$ is the indicator function. S is the set of all slot types, $ER_{s_i} = [er_1, er_2, \ldots, er_q]^T$ is the ER feature vector and $W_{s_i} = [w_{s_i 1}, w_{s_i 2}, \ldots, w_{s_i p}]$ is the corresponding weight vector. If an utterance in Music only contains SongName slot s_1, then $y = W_G G + W_{s_1} ER_{s_1}$, the rest of the terms are all 0s. If an utterance does not have any ER features from all the defined slot types, $y = W_G G + w_d$. w_d serves as the default ER feature value to the reranker

when no corresponding ER features are found in the gazetteer described above. Its value is also learned during the model training.

3.4 Loss Function

We use SemER (Semantic Error Rate) (Su et al., 2018) to evaluate NLU performance. For a hypothesis, SemER is defined as E/T, where E is the total number of substitution, insertion, deletion errors of the slots, T is the total number of slots.

One choice of the loss function is the combination of expected SemER loss and expected cross entropy loss (Su et al., 2018). The loss function L_u of an utterance is defined as:

$$L_u = k_1 S_u + k_2 C_u \tag{3}$$

where S_u is the expected SemER loss: $S_u = \sum_i^N p_i \times SemER_i$, and C_u is the expected cross entropy loss: $C_u = \sum_i^N p_i \times [-t_i \log r_i - (1 - t_i) \log(1 - r_i)]$, where $r_i = \frac{1}{1+e^{-y_i}}$, $p_i = \frac{e^{y_i}}{\sum_j^5 e^{y_j}}$, $t_i = (SemER_i == 0)$, N is the number of hypotheses in utterance u.

Since our analysis showed that *ER_NO_MATCH* is a stronger signal and we expect the top hypothesis to get ER hits, we add a penalty term N_u to the loss function to penalize the loss when the 1-best hypothesis gets *ER_NO_MATCH*.

Let $r_j = max_i(r_i)$ be the best score in the current training step, and j the index for the current best hypothesis. Then no match loss term is defined as:

$$N_u = -e_j \times \log(1 - r_j) \tag{4}$$

where $e_i = \frac{\#(slot_{er\_no\_match})}{\#(slot)}$. It is the ratio of the slots with *ER_NO_MATCH* to all the slots in the i^{th} hypothesis, and if no slot gets *ER_NO_MATCH*, the loss term is zero. Then the overall loss function is updated as:

$$L_u = k_1 S_u + k_2 C_u + k_3 N_u \tag{5}$$

N_u will penalize more the hypothesis that has a high score but gets no ER hits. $k_{1,2,3}$ are the hyperparameters, L_u is the final loss term for NLU-ER Reranker.

In our experiments, we observed that the weights are higher for the ER no match feature, and the model with the new loss term had a better performance under in-domain setup, which is as expected. Also, giving higher weight to 'ER no match' decreases the confidence scores generated by a domain NLU-ER reranker, which can help with the cross domain calibration problem. We will discuss how to ensure comparable scores in the next section.

4 Score Distribution Matching

Before adding the ER features, the reranking scores are calibrated through training on the same utterance data with similar models. However, adding the ER features in NLU reranking for a single domain may lead to incomparable scores with other domains. Using the loss function in Eq (3), we have the following theorem:

Theorem 4.1. *Under the loss function in Eq (3), assuming hypothesis 1 is the ground truth, and* $0 = SemER_1 < SemER_2 < SemER_3 < SemER_4 < SemER_5$, *with a uniform score assumption* $\sum_j^5 e^{y_j} = c$; *Eq (1) will obtain a higher positive label hypothesis score and a lower negative label score than Eq (2).*

Proof. For the expected SemER loss S_u, since it is the linear combination of $SemER_i$, the solution of the minimization problem will be: $p_1 \to 1, p_2 = p_3 = p_4 = p_5 \to 0$. This leads to:$y_1 \to \infty, y_2 = y_3 = y_4 = y_5 \to -\infty$. Then for the expected cross entropy loss C_u, let $x_i = e^{y_i}$, the minimization of C_u becomes:

$$\min -x_1 \log \frac{x_1}{1 + x_1} - \sum_{j \neq 1} x_j \log \frac{1}{1 + x_j} = \min -I_1 - I_2.$$

The first part (I_1) is monotonically increasing, while the second part (I_2) is monotonically decreasing when $x_j > 0$. This also leads to: $y_1 \to \infty, y_2 = y_3 = y_4 = y_5 \to -\infty$. Thus, solving the minimization problem $minL_u$ is equivalent to solving the linear system:

$$\begin{cases} F_+ \vec{w} = y\mathbf{1}_+ \\ F_- \vec{w} = -y\mathbf{1}_- \end{cases} \tag{6}$$

when $y \to \infty$ associated with the given loss in Eq (3), where F_+ is the feature matrix for the positive labels, F_- is the feature vector for the negative labels, $\vec{w}$ is the weight vector we need to solve, and $\mathbf{1}_+, \mathbf{1}_-$ are the unit vectors with the same dimension as the number of positive samples and negative samples respectively.

We can rewrite Eq (6) into: $F\vec{w} = \vec{y}$, and its solution will be the projection associated with the loss in Eq (3) of $\vec{y}$ onto the solution space spanned by the column vectors of matrix F. Now

define this projection as $P_F(\vec{y})$. For the feature matrix of the NLU model in Eq (2), we have $F_N = G$, and for the feature matrix of NLU-ER model in Eq (1) we have $F_{ER} = [G, ER_{s_1}, ER_{s_2}, \ldots, ER_{s_q}, \mathbb{1}_{default}]$. Since F_N is the submatrix of F_{ER}, we have $spanF_N \subset spanF_{ER}$, thus:

$$P_{F_N}(\vec{y}) \leq P_{F_{ER}}(\vec{y})$$

□

In Theorem 4.1, we show that the candidate hypothesis from a more complicated model will be more likely to have a higher score than the domains using the original reranker model. Thus the domains using the NLU-ER reranker are no longer comparable to the domains using the original model. We observed this scenario in our experiments empirically. When we only experiment with Music domain, it will generate higher confidence scores and have more false positives.

To solve this problem, since we would like the confidence scores for each domain to have stabilized variance and minimized skewness, we propose to use power transformation, which is able to map data from any distribution to an approximately standard Gaussian distribution. In our case, the confidence scores from Eq (1) might be zero or negative, thus we consider the Yeo-Johnson transformation with $\lambda \neq 0$ and $\lambda \neq 2$:

$$x_i^{(\lambda)} = \begin{cases} [(x_i+1)^\lambda - 1]/\lambda & \text{if } x_i \geq 0, \\ \frac{-[(-x_i+1)^{2-\lambda}-1]}{2-\lambda} & \text{if } x_i < 0, \end{cases} \quad (7)$$

We have the inverse function:

$$x_i^{(\lambda)} = \begin{cases} (\lambda x_i + 1)^{\frac{1}{\lambda}} - 1 & \text{if } x_i \geq 0, \\ 1 - [1-(2-\lambda)x_i]^{\frac{1}{2-\lambda}} & \text{if } x_i < 0, \end{cases} \quad (8)$$

where parameter λ is determined through maximum likelihood estimation. The idea is to first map both the NLU reranker model scores and the NLU-ER reranker scores to a standard Gaussian distribution and obtain λ_{NLU} and λ_{NLU-ER}. Then to calculate a new score from the NLU-ER reranker, we first use Eq (7) to transform the score into a standard Gaussian score with $\lambda = \lambda_{NLU-ER}$, followed by Eq (8) to transform the standard Gaussian score back into the original NLU reranker scores with $\lambda = \lambda_{NLU}$. Notice that when $\lambda > 0$, both Eq (7) and (8) are monotonic functions, thus the mapping method can only change the score distribution while maintaining the in-domain ranking order.

5 Experiment

5.1 Experimental Setup

We use the following data sets for training and evaluation:

Annotation Data (AD): It contains around 1 million annotated utterances from internal traffic. Training and testing split is 50:50. For testing, we further evaluate two different conditions: (i) 'AD All' using utterances from all domains for cross-domain evaluation. (ii) 'AD Music', 'AD Video', 'AD LS' using utterances from the Music domain, Video Domain and Local Search domain, respectively, for in-domain evaluation.

Synthetic Data (SD): These are synthetically generated ambiguous utterances used for in-domain evaluation. For Music and Video domains, utterances are in the form of "play X". Slot type of X could be ArtistName, SongName, AlbumName, VideoName, etc. X is an actual entity sampled from the corresponding ER song, video, artist, or album catalogs, and it is not in the training data, such that the model cannot infer the slot by simply "memorizing" it from the training data. We only report SongName (10K data) results in Music domain, and VideoName results in Video domain, due to the space limitation. For Local Search domain, utterances are in the form of "give me the direction to X", slot type of X could be PlaceName, DestinationName, etc. Note this data set is more ambiguous than the above one from real traffic in that "X" has multiple interpretations, whereas in real traffic users often add other words to help disambiguate, for example 'play music ...'.

We initialize the general feature weights to the same weights used in the baseline model. ER feature weights are set to smaller values (3 times less than the general feature weights). We find the expected SemER loss is less effective, so we set k_1 = 0.01, k_2 = 0.9, k_3 = 0.1. Besides, we use Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001 and train the model for 10 epochs.

5.2 Results

Table 1 presents the NLU-ER reranker results for cross-domain (AD All) and in-domain (AD Music, SD) settings. All the results are the SemER metric relative improvements compared to the baseline reranker. We have DC, IC, NER scores as the general NLU features. NLU-ER reranker uses additional ER features: ER success, no match, and lexical similarity of different slot types, and the

Table 1: NLU-ER reranking results on different data sets. The reported numbers show relative improvements compared with the baseline model using SemER evaluation metric. Baseline: NLU reranker with general features; ER: NLU-ER reranker with gazetteer selection; +N: with loss term for No Match feature; +R: with regression score matching; +P: with power transformation score matching. All the results in the table are statistically significant with p-value < 0.01.

	ER	ER+N	ER+N+R	ER+N+P
AD All	-0.22%	+0.19%	+0.26%	+0.32%
AD Music	+0.87%	+0.99%	+0.99%	+0.99%
AD Video	+0.95%	+1.01%	+1.01%	+1.01%
AD LS	+0.08%	+0.09%	+0.09%	+0.09%
SD Music	+20.74%	+28.58%	+28.58%	+28.58%
SD Video	+14.21%	+18.69%	+18.69%	+18.69%
SD LS	+12.53%	+17.37%	+17.37%	+17.37%

gazetteer selection algorithm is applied to retrieve the ER features. For the in-domain results, NLU-ER reranker has statistically significant improvement on both AD and SD. The improvement is more substantial on SD data, over 20%, which indicates ER features are more helpful when the utterances have ambiguity. Note there is some degradation in cross domain results on AD All when NLU-ER is used, due to the non-comparable score issue. After adding the loss term for ER no match feature, we observed additional improvements on both the in-domain and cross-domain settings.

As discussed earlier, because the scores from the baseline model are already well calibrated across domains, we use Yeo-Johnson transformation to match the domain score distribution back into the baseline score distribution. For Music domain, we use maximum likelihood estimation to get $\lambda_{NLU} = 1.088$ and $\lambda_{NLUER} = 1.104$. With these two estimations, we map NLU-ER reranker scores back to obtain a score in the baseline reranker score distribution. Using this updated score, we can see the cross-domain SemER score is improved by 0.32% relatively. Among the improved cases, we found that the number of False Positive utterances is decreased by 7.37% relatively. For comparison, we also trained a univariate neural network regression model to predict the original reranker score given the NLU-ER reranker score. Although this method also yields improvements, we can see that power transformation has a better performance and is also easy to implement. Note again that the in-domain performance remains the same since these score mapping approaches do not affect the in-domain ranking order. We perform the same experiments for Video domain and Local Search domain as well, and have the similar observations.

To illustrate the effectiveness of our proposed NLU-ER reranker and analyze the reasons for performance improvement, we compare the generated 1-best hypothesis from the baseline model with our new reranker. For utterance "play hot chocolate by polar express", the correct type for "polar express" is album. The baseline model predicts "polar express" as an artist because it is not in the training set, and "Song by Artist" appears more frequently than "Song by Album". However, our model successfully selected this hypothesis ("polar express" is an album), since *ER_SUCCESS* signal is found from the ER album catalog but *ER_NO_MATCH* is found from ER artist catalog. Similarly, in another example "play a sixteen z" where "a sixteen z" is ambiguous and not in the training set, the baseline model predicts it as a song since utterances with SongName slot have higher frequency in the training data, whereas our model can correctly select ProgramName as the 1-best hypothesis using ER signals.

6 Conclusion

In this work, we proposed a framework to incorporate ER information in NLU reranking. We developed a new feature vector for the domain reranker by utilizing entity resolution features such as hits or no hits. To provide the ER features to the NLU reranker, we proposed an offline solution that distills the ER signals into a gazetteer. We also introduced a novel loss term based on ER signals to discourage the domain reranker from promoting hypotheses with ER no match and showed that it leads to better model performance. Finally, since domain rerankers predict the ranking scores independently, we introduced a score matching method to transform the NLU-ER model score distribution to make the final scores comparable across domains. Our experiments demonstrated that the Music domain reranker performance is significantly improved when ER information is incorporated in the feature vector. Also with score calibration, we achieve moderate gain for the cross-domain scenario.

7 Acknowledgement

We acknowledge Kyle Saggar, Grace Deng, Ian Gardiner, Mark Cusick, Justin Flammia, Apoorva

Balevalachilu, Huitian Lei, Tan Xiao, Tian Zhu, Prajit Reddy Muppidi, Priya Khokher, Adi Gollapudi, Bo Xiao for their contributions to this effort.

References

Christopher J. Burges, Robert Ragno, and Quoc V. Le. 2007. Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 193–200. MIT Press.

Paul A Crook, Jean-Philippe Robichaud, and Ruhi Sarikaya. 2015. Multi-language hypotheses ranking and domain tracking for open domain dialogue systems. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Jean-Philippe Robichaud, Paul A Crook, Puyang Xu, Omar Zia Khan, and Ruhi Sarikaya. 2014. Hypotheses ranking for robust domain classification and tracking in dialogue systems. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–676. IEEE.

Daniel S Wilks. 1990. On the combination of forecast probabilities for consecutive precipitation periods. *Weather and forecasting*, 5(4):640–650.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.