Learning to Rank Intents in Voice Assistants

Raviteja Anantha, Srinivas Chappidi, and William Dawoodi

Abstract Voice Assistants aim to fulfill user requests by choosing the best intent from multiple options generated by its Automated Speech Recognition and Natural Language Understanding sub-systems. However, voice assistants do not always produce the expected results. This can happen because voice assistants choose from ambiguous intents—user-specific or domain-specific contextual information reduces the ambiguity of the user request. Additionally the user information-state can be leveraged to understand how relevant/executable a specific intent is for a user request. In this work, we propose a novel Energy-based model for the intent ranking task, where we learn an affinity metric and model the trade-off between extracted meaning from speech utterances and relevance/executability aspects of the intent. Furthermore we present a Multisource Denoising Autoencoder based pretraining that is capable of learning fused representations of data from multiple sources. We empirically show our approach outperforms existing state of the art methods by reducing the error-rate by 3.8%, which in turn reduces ambiguity and eliminates undesired dead-ends leading to better user experience. Finally, we evaluate the robustness of our algorithm on the intent ranking task and show our algorithm improves the robustness by 33.3%.

1 Introduction

A variety of tasks use Voice Assistants (VA) as their main user interface. VAs must overcome complex problems and hence they typically are formed of a number of components: one that transcribes the user speech (Automated Speech Recognition - ASR), one that understands the transcribed utterances (Natural Language Understanding - NLU), one that makes decisions (Decision Making - DM [24]), and one

R. Anantha (✉) · S. Chappidi · W. Dawoodi
Apple Inc., Seattle, USA
e-mail: raviteja_anantha@apple.com

S. Chappidi
e-mail: vasuc@apple.com

W. Dawoodi
e-mail: dawoodi@apple.com

L. F. D'Haro et al. (eds.), *Conversational Dialogue Systems for the Next Decade*, Lecture Notes in Electrical Engineering 704, https://doi.org/10.1007/978-981-15-8395-7_7

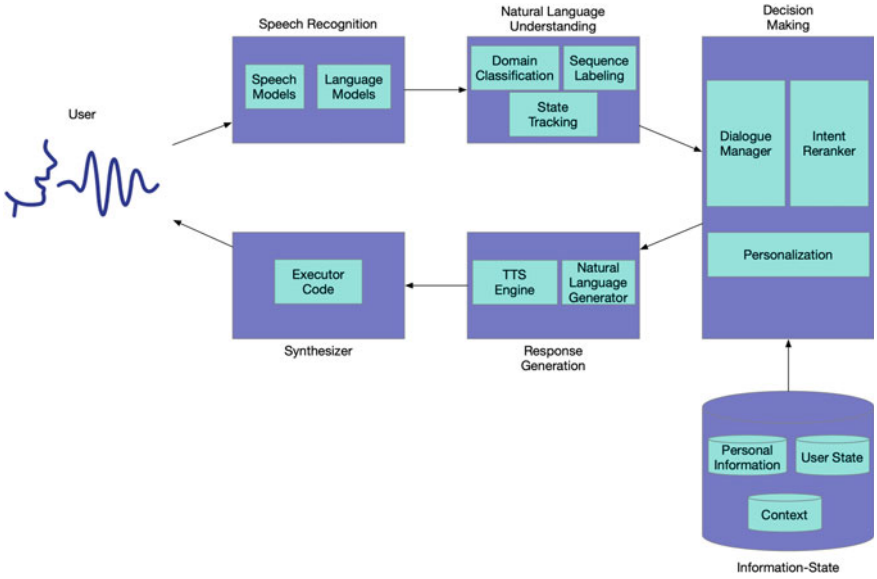

Fig. 1 Components of a voice assistant

that produces the output speech (TTS). Many VAs have a pipeline structure similar to that in Fig. 1.

Our work is mainly focused on the DM sub-system and our primary contributions are: (1) proposing to decouple language understanding from information-state and modeling an affinity metric between them; (2) the identification of Multi-source Denoising Autoencoder based pretraining and its application to learn robust fused representations; (3) quantifying robustness; (4) the introduction of a novel ranking algorithm using Energy-based models (EBMs). In this work, we limit our scope to non-conversational utterances, *i.e.*, utterances without followups containing anaphoric references and leave that for future work. We evaluate our approach on an internal dataset. Since our algorithm is primarily focused on leveraging inherent characteristics that are unique to large-scale real-world VAs, the exact algorithm may not be directly applicable to open-source *Learning to Rank* (LTR) datasets. But we hope our findings will encourage application and exploration of EBMs applied to LTR in both real-world VAs and other LTR settings.

The remainder of the paper is organized as follows: Sect. 2 discusses the task description while Sect. 3 covers the related work. Section 4 then describes the ranking algorithm, and Sect. 5 discusses the evaluation metrics, datasets, training procedure, and results.

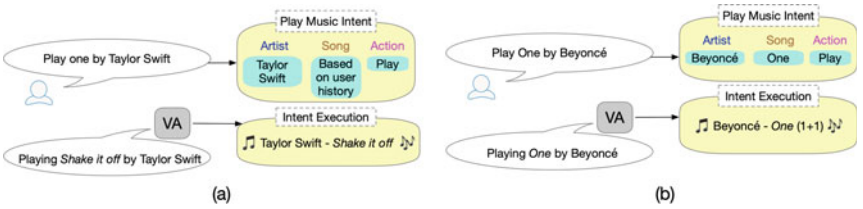

Fig. 2 Examples of user requests with same semantics but with different intents. **(a)** shows a user request to play a song from an artist, **(b)** shows a user request to play a specific song from an artist

2 Task Description

The ultimate goal of a VA is to understand user intent. The exact meaning of the words is often not enough to choose the best intent. In Fig. 1, we show the use of information-state, and we classify it into three categories. All private-sensitive information stays on the user's device.

Personal Information: *e.g.* user-location, app subscriptions, browsing history, device-type etc.

User State: Information about the user's state at the time a query is made. (*e.g.* user is driving, etc.)

Context: Dialog context of what the user said in previous queries in the same conversation or task (*e.g.* song requests).

To illustrate how semantically similar user requests can have different user intents consider the examples in Fig. 2. In Fig. 2a the user meant to play some song from a specific artist. However in Fig. 2b, although playing some song from the requested artist is also reasonable, knowing that there is a song named "One" from the artist leads to better intent selection, as shown.

Ambiguity can still remain even if a sub-system correctly decodes user input. For example consider Fig. 3: it is not possible to predict the user intended transcription unless we know there is a contact with that name due to the homophone. Figure 3b is an example where a suboptimal intent was executed although there was a better intent as shown in Fig. 3c. We term this scenario *undesired dead-end* since the user's intended task hit a dead-end.

The use of information-state is crucial to select the right response, which is also shown empirically in Sect. 5.4.1. We aim to reduce ambiguity (both ASR and NLU), and undesired dead-ends to improve the selection of the right intent by ranking alternative intents. ASR signals are comprised of speech and language features that generate speech lattices, model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described

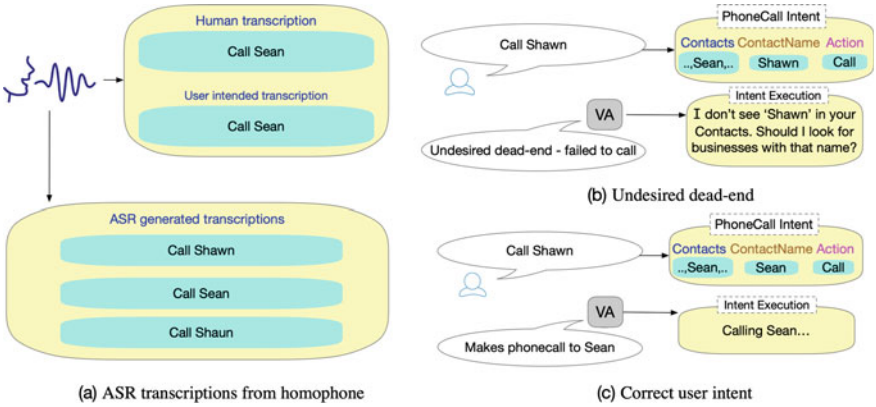

Fig. 3 An example of an undesired dead-end. **(a)** shows a case where user intended transcription is not possible to predict unless the voice assistant has the contact information. **(b)** shows how lack of contact information leads to a sub-optimal intent execution although there is a better intent shown in **(c)**

in Sect. 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response.

3 Related Work

While our work falls within the broad literature of LTR, we position it in the context of information-state based reranking, unsupervised pretraining, zero-shot learning, and EBMs applied to the DM sub-system of a Voice Assistant.

Information-State Based Reranking: Reranking approaches have been used in VAs to rerank intents to improve accuracy. Response category classification can be improved by reranking k-best transcriptions from multiple ASR engines [18]. ASR accuracy can be improved by reranking multiple ASR candidates by using their syntactic properties in Human-Computer Interaction [1]. Reranking domain hypotheses is shown to improve domain classification accuracy over just using domain classifiers without reranking [13, 20].

All of the above approaches only focus on ASR candidates or domain hypotheses, which are strongly biased towards the semantics of the user request. Although [13] exploits user preferences along with NLU interpretation, they treat both of them as a single entity (hypothesis). In our work, we explicitly learn an affinity metric between information-state and predicted meaning from the transcribed utterance to choose the appropriate response.

Unsupervised Pretraining: DM input consists of multiple diverse sources. For example, speech lattices, textual information, scores from ASR and NLU models,

and unstructured contextual information, to name a few. Each data type has distinct characteristics, and learning representations across data types that capture the meaning of the user request is important. One approach is to use a deep boltzmann machine for learning a generative model to encode such multisource features [22]. Few approaches learn initial representations from unlabeled data through pretraining [1, 20]. Encoding can also be learned by optimizing a neural network classifier weights by minimizing the combined loss of an autoencoder and a classifier [19]. Both pretraining and classification can be jointly learned from labeled and unlabeled data, labeled data loss is used to obtain pseudo-labels, and pretraining is done using the pseudo-loss [17]. Pretraining for initial representations can also be realized by using a CNN2CRF architecture for slot tagging using labeled data, and learning dependencies both within and between latent clusters of unseen words [6].

Although these previous works address few aspects of the multisource data problem, none of them address the robustness of the learned representations. Since DM consumes the outputs of many sub-systems that may change their distributional properties, for instance through retraining, some degree of robustness is desired to not drastically affect the response selection.

To address both distinct data characteristics and robustness, we propose using a Denoising Autoencoder (DAE) [25] with a hierarchical topology that uses separate encoders for each data type. The average reconstruction loss contains both a separate term to minimize the error for each encoder, and the fused representations. This provides an unsupervised method for learning meaningful underlying fused representations of the multisource input.

Zero-Shot Learning: The ability of DM to predict and select unseen intents is important. User requests can consist of word sequences that NLU might not be able to accurately tag by relying only on language features. To illustrate consider the examples in Fig. 4. The user request in Fig. 4a is tagged correctly, and the NLU subsystem predicts the right user intent of playing a song from the correct artist. Figure 4b showcases a scenario where due to external noise the user intended transcription of "Play ME by Taylor Swift" was mistranscribed by the ASR sub-system as "Play me Taylor Swift", and this ASR error propagated to NLU leading to tag *ME* as a pronoun instead of *MusicTitle*. With DM, as shown in Fig. 4c, we leverage domain-specific information and decode the right transcription and intent (playing ME song) from the affinity metric, although this input combination was never seen before by the model.

One approach is to use a convolutional deep structured semantic model (CDSSM), which performs zero-shot learning by jointly learning the representations for user intents and associated utterances [7]. This approach is not scalable since such queries can have numerous variations, and they follow no semantic pattern. We propose to complement NLU features with domain-specific information to decode the right intent in addition to shared semantic signals.

EBM for DM: Traditional approaches to LTR use discriminative methods. Our approach learns an affinity metric that captures dependencies and correlations between semantics and information-state of the user request. We accomplish this

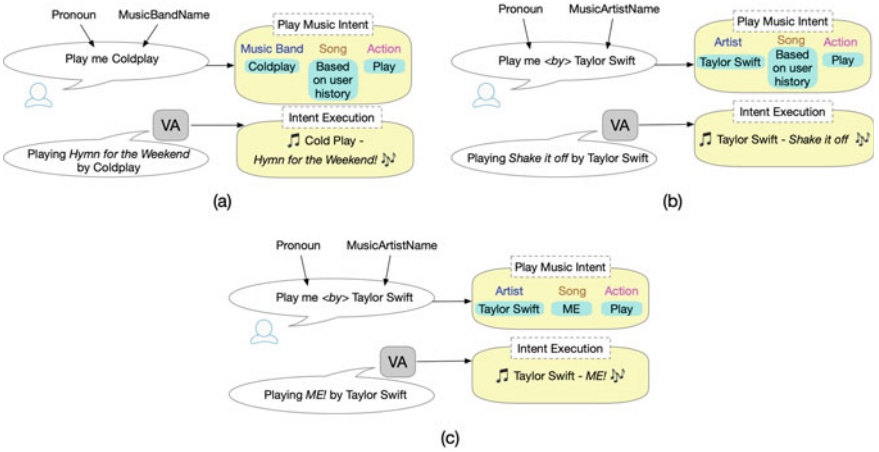

Fig. 4 **(a)** Shows a scenario where NLU correctly predicts the intent given correct ASR transcription. **(b)** Shows a scenario where NLU fails to predict the right intent due to incorrect ASR transcription (missing the word "by") caused by external noise. **(c)** Shows a scenario where NLU fails to predict the right intent, but DM helps in identifying the correct intent using domain-specific information

learning by associating a scalar energy (a measure of compatibility) to each configuration of the model parameters. This learning framework is known as *energy-based learning* and is used in various computer vision applications, such as signature verification [2], face verification [9], and one-shot image recognition [15]. We apply EBM for LTR (and DM in voice assistants) for the first time. We propose a novel energy-based learning ranking loss function.

4 EnergyRank Algorithm

EBMs assign unnormalized energy to all possible configurations of the variables [16, 23]. The advantage of EBMs over traditional probabilistic models, especially generative models, is that there is no need for estimating normalized probability distributions over the input space. This is efficient since we can avoid computing partition functions. Our algorithm consists of two phases—pretraining and learning the ranking function, which are described in Sects. 4.1 and 4.2 respectively.

4.1 Multisource Denoising Autoencoder

Since our model consumes input from multiple sub-systems, two aspects are important: robustness of features and efficient encoding of multisource input. The concept

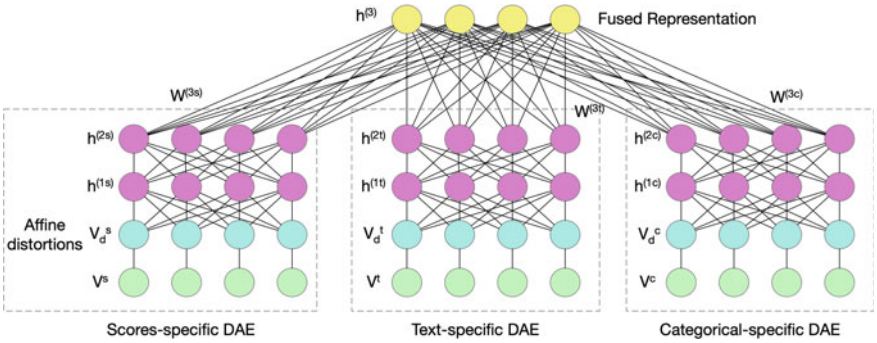

Fig. 5 Encoder architecture of Multisource DAE that models the joint distribution over scores, text, and categorical features. *Light green* layer, V^*, represents the original input; *light magenta* layer, V_d^*, depicts the affine transformations; two layers of *dark magenta*, h^{1*} and h^{2*}, represents source-specific latent representation learning; finally, *light yellow* layer, $h^{(3)}$, represents the fused representation

of DAE [25] is to be robust to variations of the input. We have three data types in the input: model scores that are produced by other sub-systems, text generated by ASR and Language Models (LMs), categorical features generated by NLU models like sequence labels, verbs etc. Let V^s denote a multi-hot vector, which is a concatenation of 11 $\mathbb{R}^{11}$ one-hot vectors, where each contains binned real-valued model scores. Let V^t represent the associated text input (padded or trimmed to a maximum of 20 words), which is a concatenation of 20 word-vectors. Each word-vector $v_i^t \in \mathbb{R}^{50}$ is a multi-hot vector of i^{th} word. Similarly let V^c represent associated sequence-labels of those 20 words, which is a concatenation of 20 sequence-label vectors. Each i^{th} sequence-label vector $v_i^c \in \mathbb{R}^{50}$ is a multi-hot vector. For example consider the utterance "Call Ravi", the corresponding sequence-labels might be [*phoneCallVerb*, *contactName*].

We start by modeling each data type by adding affine distortions followed by a separate two-layer projection of the encoder, as shown in Fig. 5. This gives separate encodings for each data type. Let dae_* represent an encoding function, W_{enc}^* is the respective weight matrix and $P(noise)$ a uniform noise distribution. The encodings are given by:

$$V_d^s, V_d^t, V_d^c = affine\_transform((V^s, V^t, V^c); P(noise)). \tag{1}$$

Let us denote source-specific hidden representations of real-valued, text and categorical features by h^s, h^t, h^c derived from encoder models with respective parameters W_{enc}^s, W_{enc}^t, W_{enc}^c. These latent representations are given by:

$$h^* = dae_*(V_d^*; W_{enc}^*), \tag{2}$$

and the fused representation is obtained by:

$$h = dae((h^s, h^t, h^c); W_{enc}). \tag{3}$$

Let $idae_*$ represent the decoding function, and W^*_{dec} denote the respective weight matrix. The hidden-state reconstructions are given by:

$$h^{s'}, h^{t'}, h^{c'} = idae(h; (W^{s'}_{dec}, W^{t'}_{dec}, W^{c'}_{dec})). \tag{4}$$

The original denoised input reconstructions are given by:

$$V^{*'} = idae_*(h^{*'}; W^*_{dec}). \tag{5}$$

We learn the parameters of the Multisource DAE jointly by minimizing the average reconstruction error captured by *categorical cross entropy* (CCE) of both the hidden state and the original denoised input decodings captured by the terms of the loss function. We denote the CCE loss as L_{CCE}.

$$L^h = L_{CCE}(h^*, h^{*'}), \tag{6}$$

$$L^V = L_{CCE}(V^*, V^{*'}), \tag{7}$$

$$W^*_{enc}, W_{enc}, W^*_{dec} = \underset{W^*_{enc}, W^*_{dec}}{\arg\min} \frac{1}{m} \sum_{i=1}^{m} (L^h_i + L^V_i). \tag{8}$$

4.2 Model Description

The ranking function is learned by finding the parameters W that optimize the suitably designed ranking loss function evaluated over a validation set. Directly optimizing the loss averaged over an epoch generally leads to unstable EBM training and would be unlikely to converge [9]. Therefore, we add a *scoring layer* after the energy is computed and impose loss function forms to implicitly ensure energy is large for intent with bad rank and low otherwise. Details of the energy computation and the loss function forms are given in Sects. 4.2.1 and 4.2.2 respectively.

4.2.1 Energy Function of EBM

The architecture of our Ranker is shown in Fig. 6. Our ranker consists of two identical Bidirectional RNN networks, where one network accepts the fused representation, and the other accepts the information-state. Learning the affinity metric is realized by training these twin networks with shared weights. This type of architecture is called a Siamese Network [2]. The major difference between our work and previous works on siamese networks is that we present the same data-point to the twin networks

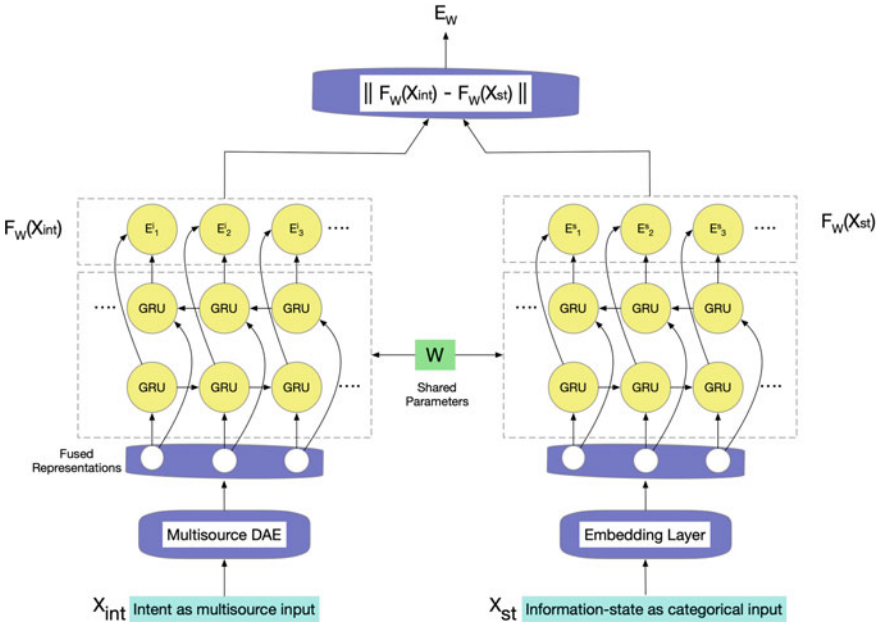

Fig. 6 EBM with siamese architecture

but categorized as two inputs based on if it is information-state or not. All previous works use two distinct data-points to compute energy. In other words, we compute intra-energy and previous works focused on inter-energy. We used GRU [8] for the RNN since it often has the same capacity as an LSTM [11], but with fewer parameters to train.

To simplify let X_{int} and X_{st} denote an intent's extracted meaning (V^s, V^t, V^c) and its associated information-state respectively. Both the inputs are transformed through Multisource DAE and Embeddings Layer respectively to have the same dimensions $\mathbb{R}^{500}$. Let W be the shared parameter matrix that is subject to learning, and let $F_W(X_{int})$ and $F_W(X_{st})$ be the two points in the metric space that are generated by mapping X_{int} and X_{st}. The parameter matrix W is shared even if the data sources of X_{int} and X_{st} are different since they are related to the same request and the model must learn the affinity between them. We compute the distance between $F_W(X_{int})$ and $F_W(X_{st})$ using the L1 norm, then the energy function that measures compatibility between X_{int} and X_{st} is defined as:

$$E_W(X_{int}, X_{st}) = \|F_W(X_{int}) - F_W(X_{st})\|. \tag{9}$$

4.2.2 Energy-Based Ranking Loss Function

Traditional ranking loss functions construct the loss using some form of entropy in a pointwise, pairwise or listwise paradigm. Parameter updates are performed using either *gradients* [3] or *Lambdas* λ [4, 5]. We use gradient based methods to update parameters. Let x_1 and x_2 be two intents from same user request. The prediction score of the ranker is obtained by $p = \sigma(E_W)$, for convenience we denote p associated with x_1 as $p(x_1)$ and $f(.)$ as the learned model function. We construct the loss as a sequence of weighted energy scores. Pairwise loss is constructed as:

$$L(f(.), x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \phi(p(x_i), p(x_j)), \tag{10}$$

where ϕ is a hyperparameter that can be one of logistic function ($\phi(z) = log(1 + \exp^{-z})$), hinge function ($\phi(z) = (1 - z)_+$), exponential function ($\phi(z) = \exp^{-z}$), with $z = p(x_i) - p(x_j)$.

Listwise losses are constructed as:

$$L(p(.), x, y) = \sum_{i=1}^{n-1}(-p(x_{y(s)}) + ln(\sum_{j=i}^{n} \exp(p(x_{y(i)})))), \tag{11}$$

where y is a randomly selected permutation from the list of all possible intents that retains relevance to the user-request.

5 Experiments and Results

5.1 Evaluation Metrics

We evaluated EnergyRank using two metrics.

- **Error Rate**: The fraction of user requests where the intent selection was incorrect.
- **Relative Entropy**: We employ *Relative Entropy*, given in Eq. 12, to quantify the distance between input score distributions p and q. Relative entropy serves as a measure for the robustness of the model to upstream sub-system changes. We used *whitening* to eliminate unbounded values, and 10E−9 as a dampening factor to give a bounded metric. A value of 0.0 indicates identical distributions, while 1.0 are maximally dissimilar.

$$rel\_entr(p, q) = \begin{cases} p \log(p/q) & p > 0, q > 0 \\ 0 & p = 0, q \geq 0 \\ \infty & otherwise. \end{cases} \tag{12}$$

5.2 Datasets

5.2.1 Labeled Dataset

The labeled dataset is used to measure the error rate. This dataset contains 24,000 user requests comprised of seven domains: music, movies, app-launch, phone-call, and three knowledge-related domains. The ranking labels are produced by human annotators by taking non-private information-state into account. The dataset is divided into 12,000 user requests for training, 4,000 for validation and 8,000 for the test-set. The average number of predicted intents per user request is 9 with a maximum of 43. The extracted meaning of the request is represented by features from ASR and NLU sub-systems, information-state is represented by 114 categorical attributes. The error rate with just selecting the top hypothesis is 41%.

5.2.2 Unlabeled Dataset

The unlabeled dataset consists of two unlabeled sub-datasets sampled from two different input distributions. Each sub-dataset consists of 80,000 user requests. The data here are not annotated since we are interested in a metric that only needs the scores of the model's best intent.

5.3 Training Procedure

We trained EBM using both pairwise and listwise loss functions given in Eqs. 10 and 11 respectively. The objective is combined with backpropagation, where the gradient is additive across the twin networks due to the shared parameters. We used a minibatch size of 32 and Adam [14] optimizer with the default parameters. For regularization, we observed that Batch Normalization [12] provided better results than Dropout [21].

We used *tanh* for GRU and *ReLU* for all units as activation functions. We initialized all network weights from a normal distribution with variance $2.0/n$ [10], where n is the number of units in previous layer. Although we use an adaptive optimizer, employing an exponential decay learning schedule helped improve performance. We trained EBM for a maximum of 150 epochs.

5.4 Results

We trained three baseline algorithms: Logistic Regression, LambdaMART [4], and HypRank [13], where Logistic Regression and LambdaMART were trained with

Table 1 Error-rates on labeled data both with and without information-state.

Method	Error rate*	p-value*	Error rate**	p-value**
$LogisticRegression$	41.1% ± 0.5%	0.7E − 04	32.1% ± 1.2%	1.2E − 05
$LambdaMART^{OH}$	36.5% ± 0.3%	1.4E − 05	22.3% ± 0.1%	1.1E − 05
$EnergyRank^{EF}_{list}$	—	—	20.9% ± 1.3%	0.9E − 05
$LambdaMART^{FH}$	34.4% ± 0.6%	1.3E − 05	20.2% ± 0.1%	1.1E − 05
$HypRank$	32.9% ± 0.8%	1.6E − 04	19.6% ± 0.9%	2.3E − 04
$EnergyRank^{HF}_{pair}$	—	—	19.5% ± 0.6%	1.6E − 03
$LambdaMART^{ED}$	**29.7% ± 0.3%**	0.9E − 05	18.2% ± 0.1%	1.2E − 05
$EnergyRank^{LF}_{list}$	—	—	17.9% ± 1.1%	2.1E − 03
$EnergyRank^{LF}_{pair}$	—	—	**17.5% ± 0.8%**	1.3E − 05

* without information-state
** with information-state

the pairwise loss function, HypRank with the listwise loss function, and EnergyRank with both loss functions. For LambdaMART we used three different encoding schemes: one-hot vectors (OH), feature hashing (FH), and eigen-decomposition (ED). For HypRank we used $LSTM^C$, i.e, concatenating the hypothesis vectors and the BiLSTM output vectors as input to the feedforward layer since this was the best performing architecture.

5.4.1 Error Rate

We trained each model ten times with different seed and weight initializations, and we report the mean error rate. We use a two-sided T-test to compute p-value to establish statistical significance. Table 1 shows the results on the internal labeled dataset, with ± showing 95% confidence intervals. We empirically show that information-state improves error-rates. EnergyRank results are not reported in experiments without information-state since it needs both understanding features and information-state to compute the affinity metric. The superscript of LambdaMART denotes the encoding scheme used. EnergyRank superscript denotes ϕ used: EF for Exponential Function, HF for Hinge Function, LF for Logistic Function, and subscript for pairwise/listwise loss paradigm.

5.4.2 Relative Entropy

We run the best performing methods: LambdaMART, HypRank, and EnergyRank models on two unlabeled datatsets, each of the size 80,000 sampled from different feature distributions. We use the score of the model's top predicted intent and group them into 21 buckets ranging from 0.0 to 1.0 with a step-size of 0.05. The raw counts obtained are normalized and interpolated to obtain a probability

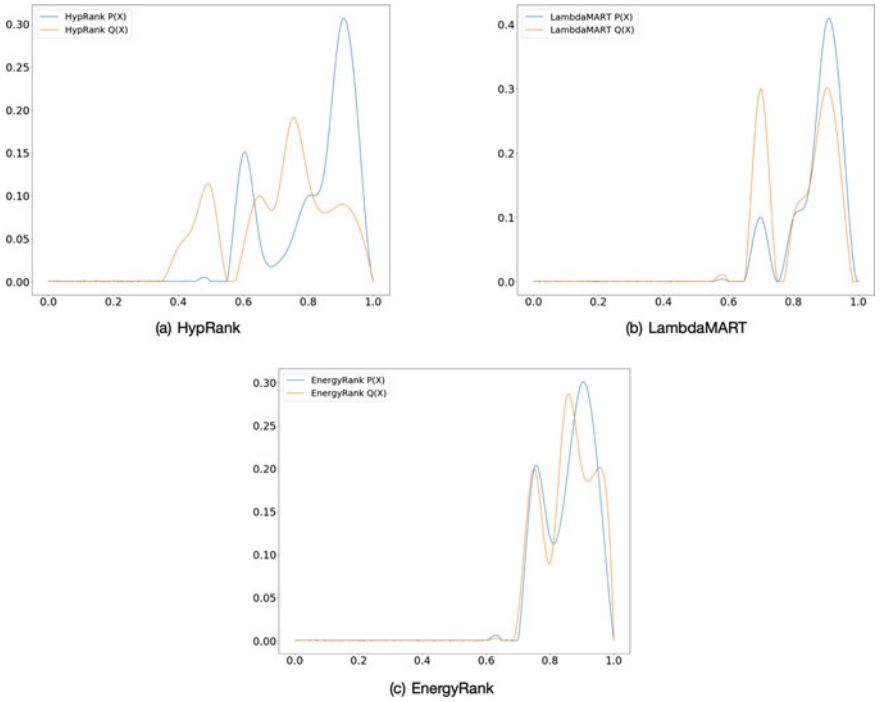

Fig. 7 A visualization of the model's top intent score distributions as probability density function (PDF) corresponding to two different input distributions P(X) and Q(X)

Table 2 Relative-entropies on unlabeled data.

Method	Relative entropy
$HypRank$	0.468
$EnergyRank_{pair}^{LF-NA}$	0.319
$LambdaMART^{ED}$	0.168
$EnergyRank_{pair}^{LF}$	**0.112**

density function (PDF) of the scores. We measure the relative entropy to quantify the robustness of these algorithms to changes in feature distributions. The best performing EnergyRank model degrades in robustness when no affine-transform is applied ($EnergyRank_{pair}^{LF-NA}$) with a minimal drop in accuracy.

Figures 7a, b, and c show the superimposition of the model's top intent output score PDFs of HypRank, LambdaMART, and EnergyRank respectively. The two output score PDFs in each superimposition correspond to P(X) and Q(X) input distributions. Table 2 shows the relative-entropy which quantifies the difference between the two PDFs. EnergyRank with pairwise loss improves relative-entropy over LambdaMART with ED (best performing method among SOTAs, see Table 1) by 33.3% and over HypRank by 76.1%.

6 Conclusion

We have presented a novel ranking algorithm based on EBM for learning complex affinity metrics between extracted meaning from user requests and user information-state to choose the best response in a voice assistant. We described a Multisource DAE pretraining approach to obtain robust fused representations of data from different sources. We illustrated how our model is also capable of performing zero-shot decision making for predicting and selecting intents. We further evaluated our model against other SOTA methods for robustness and show our approach improves relative-entropy.

References

1. Basili R, Bastianelli E, Castellucci G, Nardi D, Perera V (2013) Kernel-based discriminative re-ranking for spoken command understanding in HRI. Springer International Publishing, Cham, pp 169–180
2. Bromley J, Guyon I, LeCun Y, Sackinger E, Shah R (1993) Signature verification using a siamese time delay neural networks. In: Advances in neural information processing systems
3. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G (2005) Learning to rank using gradient descent. In: Proceedings of international conference on machine learning
4. Christopher JCB (2010) From ranknet to lambdarank to lambdamart: an overview
5. Christopher JCB, Ragno R, Viet Le Q (2006) Learning to rank with non smooth cost functions. In: Proceedings of the NIPS
6. Celikyilmaz A, Sarikaya R, Hakkani Tur D, Liu X, Ramesh N, Tur G (2016) A new pre-training method for training deep learning models with application to spoken language understanding
7. Nung Chen Y, Hakkani Tur D, He X (2016) Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In: IEEE international conference on acoustics, speech and signal processing (ICASSP)
8. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1724–1734
9. Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. Proceeding CVPR 2005 proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR 2005), vol 1, pp 539–546
10. Kaiming H, Xiangyu Z, Shaoqing R, Jian S (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet classification
11. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
12. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on machine learning, PMLR, vol 37, pp 448–456
13. Kim YB, Kim D, Kim JK, Sarikaya R (2018) A scalable neural shortlisting-reranking approach for large-scale domain classification in natural language understanding. In: Proceedings of NAACL-HLT, pp 16–24
14. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. In: Proceedings of the international conference on machine learning
15. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition. In: Proceedings of the 32nd international conference on machine learning, Lille, France

16. LeCun Y, Huang FJ (2005) Loss functions for discriminative training of energy-based models. AI-stats
17. Lee D-H (2013) Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Proceedings of the 25th international conference on machine learning, ICML
18. Morbini F, Audhkhasi K, Artstein R, Van Segbroeck M, Sagae K, Georgiou P, Traum DR, Narayan S (2012) A reranking approach for recognition and classification of speech input in conversational dialogue systems. In: IEEE spoken language technology workshop (SLT)
19. Ranzato MA, Szummer M (2008) Semi-supervised learning of compact document representations with deep networks. In: Proceedings of the 25th international conference on machine learning, ICML
20. Robichaud JP, Crook PA, Xu P, Khan OZ, Sarikaya R (2014) Hypotheses ranking for robust domain classification and tracking in dialogue systems
21. Nitish S, Geoffrey H, Alex K, Ilya S, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting
22. Srivastava N, Salakhutdinov R (2012) Multimodal learning with deep boltzmann machines. In: Proceedings of neural information processing systems
23. Teh YW, Welling M, Osindero S, Hinton GE (2003) Energy-based models for sparse overcomplete representations. J Mach Learn Res 4:1235–1260
24. Thomson B (2013) Statistical methods for spoken dialogue management. Springer-Verlag, London
25. Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine Learning, pp 1096–1103