Energy-Based Modelling for Dialogue State Tracking

Anh Duong Trinh †, **Robert J. Ross** †, **John D. Kelleher** ‡
† School of Computer Science
‡ Information, Communications & Entertainment Institute
Technological University Dublin, Ireland
anhduong.trinh@mydit.ie, {robert.ross, john.d.kelleher}@dit.ie

Abstract

The uncertainties of language and the complexity of dialogue contexts make accurate dialogue state tracking one of the more challenging aspects of dialogue processing. To improve state tracking quality, we argue that relationships between different aspects of dialogue state must be taken into account as they can often guide a more accurate interpretation process. To this end, we present an energy-based approach to dialogue state tracking as a structured classification task. The novelty of our approach lies in the use of an energy network on top of a deep learning architecture to explore more signal correlations between network variables including input features and output labels. We demonstrate that the energy-based approach improves the performance of a deep learning dialogue state tracker towards state-of-the-art results without the need for many of the other steps required by current state-of-the-art methods.

1 Introduction

Dialogue processing is a challenging task due to the nature of human conversations. Currently most Spoken Dialogue Systems (SDS) have a core component called the Dialogue Manager that is responsible for: (a) handling dialogue context and understanding user utterances by tracking dialogue states; and (b) generating useful contributions through the use of an appropriate dialogue policy. The dialogue manager component can be developed independently (Budzianowski et al., 2017; Su et al., 2017; Zhao and Eskenazi, 2016) or in an end-to-end dialogue fashion (Williams et al., 2017; Li et al., 2017; Serban et al., 2016). Between the two dialogue manager components, the dialogue state tracker is arguably the more challenging to perfect, as its performance depends on the quality of the speech recognition component, the complexity of natural language used by users, and even the situational context (Ross and Bateman, 2009).

Generally task-oriented dialogue systems with predefined ontologies represent dialogue states as a set of slot-value pairs, and define dialogue state tracking as a multi-task classification problem. The common deep learning approach to dialogue state tracking therefore is to develop different subsystems for the tracking of each slot – though early layers in the network will often be shared to varying degrees. While this approach has provided reasonable results, we argue that this method does not reflect the natural way that humans process information; specifically that the inter-relationships between slots are not properly taken into account.

In order to account for such relationships in the dialogue context, it is appropriate to consider the problem not as a multi-task classification problem, as is currently common, but as a structured prediction problem. This insight is not in itself novel, as there have been several attempts in the research community to investigate the variable dependencies in dialogue state tracking such as in the multi-task learning model (Trinh et al., 2018), the language modelling tracker (Platek et al., 2016), work building on Conditional Random Fields (Kim and Banchs, 2014), work on Attention-based Sequence-to-Sequence models (Hori et al., 2016) and the work by Williams (2010). Although these architectures are good attempts to engage variable dependencies at different levels of abstraction into the dialogue state tracking process, they have not yet achieved state-of-the-art results and do not provide a clear analysis of the relationships between variables.

Performing prediction of dialogue states where we acknowledge the relationship between slot values casts the problem into a structured prediction task; this is similar to how both image segmentation and part-of-speech tagging are struc-

Proceedings of the 1st Workshop on NLP for Conversational AI, pages 77–86
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

tured prediction problems in that that output labels are not assumed to be independent. One efficient approach to structured prediction that has been applied widely in recent years are energy-based methods (LeCun et al., 2006). A key intuition of energy-based structured learning approaches is that it can be easier to learn a function to critique a potential solution Y than to learn to predict Y directly from an input signal X. Given this intuition, energy-based approaches essentially attempt to learn a function that estimates the goodness of fit between some input feature variable X and an output hypothesis Y. Given such a trained function, a gradient descent-based inference process then searches for an appropriate Y at run-time that demonstrates the best fit to a new input vector X.

To investigate the appropriateness of this method, in this paper we apply a variant of the Structured Prediction Energy Network (SPEN) (Belanger and McCallum, 2016) to the Dialogue State Tracking Challenge (DSTC) 2 dataset (Henderson et al., 2014a). To our knowledge, this is the first attempt to apply this formulation of modelling to the DST task. We benchmark our work by comparing it against a number of other dialogue state trackers including the state-of-the-art hybrid dialogue state tracker (Vodolan et al., 2015, 2017).

2 Analysis of Variable Dependencies

The goal of applying a structured learning approach to dialgoue state tracking is predicated on the assumption that there are indeed dependencies between slots in the dialogue state. In this section we recap some of the features of the dataset that we have applied and investigate whether such dependencies exist for this dataset.

2.1 DSTC2 Dataset

The Dialogue State Tracking Challenge 2 (Henderson et al., 2014a) is a popular dataset for spoken dialogue state tracking in the Cambridge restaurant information domain. The main task of this challenge, called *Joint Goals*, requires the models to classify slot-value pairs for four Informable slots; namely *food*, *price range*, *area*, and *name*. At every turn of the dialogue, each slot must be assigned a value from its set of possible values detailed in the task ontology. However, the analysis shows that the slot *name* rarely appears in the dataset (see Appendix A.1). Therefore following the approach of a number of other researchers, we focus on the remaining three slots only.

The DSTC2 dataset contains 1612 dialogues in a training set, 506 in a development (validation) set, and 1117 in a test set.

2.2 Data Analysis

We conducted a data analysis on the DSTC2 data using the chi-square test to examine the dependencies between target variables. The chi-square test;is an important statistical test to detect associations between variables; however, this test can only give the answer to the question of whether there exist dependencies between variables. Therefore, it is also important to measure the strength of detected dependencies. For this purpose, we perform a chi-square test on the three informable slots in a pairwise fashion and use the chi-square test's ϕ coefficient to measure the strength of their dependencies (see Appendix A.2). The chi-square test result confirms the existence of pairwise dependencies among DSTC2 data informable slots with the statistical significance $p < 0.05$. The dependencies are reported in Table 1 with the ϕ coefficient.

	food	price	area
food	-		
price	0.608	-	
area	0.707	0.393	-

Table 1: Data analysis of variable dependencies on DSTC2 data. The result is reported with ϕ coefficient values.

The statistical test shows that there are associations of different levels among informable slots in the DSTC2 data. We observe that two pairs *food – price range* and *food – area* have strong dependencies, while the relationship *price range – area* is weaker. We argue that this observation indicates the validity of the motivation for our work in that there are dependencies between target labels and hence the dialogue state tracking task can be cast as a structured prediction problem.

3 Energy-Based Learning

Energy-Based Learning is a branch of machine learning that is notable for its usefulness in structured prediction tasks. Energy-based structured prediction methods have been applied in tasks ranging from Part-of-Speech (POS) tagging (Voutilainen, 1995; Ma and Hovy, 2016) through to instance segmentation tasks in computer vision (Corso et al., 2004; Li and Zhao, 2009; Ngiam et al., 2011). In all of these tasks the output is not a highly structured object, but is rather a set of labels that are not assumed to be independent of each other.

The main intuition behind energy-based methods is that it is too challenging to learn a structured output Y for a given input vector X, and that instead we should learn a function that essentially assesses the goodness of fit between a given structured output Y and the input vector X. In practice we often assume that the raw data is pre-processed in a domain appropriate way to give us a more useful representation of the data to evaluate against a given target. Thus the energy network actually calculates the goodness of fit between some representation of X, referenced from here on out as $F(X)$, and a candidate output Y. While in principle a wide range of methods could be used to generate a feature representation $F(X)$, in this work we assume the feature representation is generated by some form of deep network which we refer to as the feature network. For an image processing task such a network might be based on series of convolutions, while in a language processing task such a network might be based on a recurrent architecture. Given the above, we define that energy function itself simply as $E(F(X), Y)$ which returns some scalar value.

During training, an appropriate objective function $L(E, E^*)$, where $E^* = E((F)X, Y^*)$ is the ground truth energy calculated based on input feature representation $F(X)$ and target labels Y^*, is used to guide training such that the energy function is minimised for valid combinations of $F(X)$ and Y observed in the training data. During runtime we do not have gold standard values for Y, and instead we only have processed inputs $F(X)$. Thus at runtime we begin with an initial hypothesis for Y – usually that $Y = [0]^N$, and we then perform an inference process to update Y so as to find the best fit according to our learned differentiable energy function. This overall approach is illustrated by Figure 1.

The specific design of the energy function is important in achieving an appropriate estimator for goodness of fit between input vectors and candidate structured outputs. Belanger and McCallum (2016) propose an energy function based around the combination of a local and global energy where global energy gives a scalar that represents the cross correlations for the target vector Y only, and the local energy considers the relationship between the input vector X and individual elements of the total output structure variable, i.e., $y \in Y$. Both the local and global energy functions are approximated as layers in a neural network such that complex energy functions may be learned from the training data.

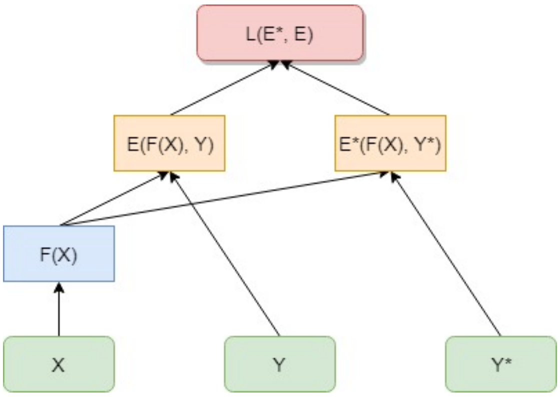

Figure 1: An example of Energy-Based Model, that consists of a feature network $F(X)$, an energy function $E(F(X), Y)$, and an objective function $L(E, E^*)$, where X is input variable, $F(X)$ is a feature representation generated by a feature network, Y is predicted output variable, and Y^* is a gold standard label output variable.

As indicated, the energy function beside being used to produce scalar energy values is also used to generate predicted output variables. This process is called the *Inference process*. Commonly a gradient-based technique is used to generate the output variable in a continuous space (Belanger and McCallum, 2016; Belanger et al., 2017). The inference process can be formulated as follow:

$$y_{t+1} \leftarrow y_t - \eta_t \nabla_y (E(X, Y)) \qquad (1)$$

where η_t is the learning rate at time step t, and $\nabla_y (E(X, Y))$ is the gradients of energy value with respect to the output variable.

The process to train the energy network parameters is called the *Learning process*, where an objective function is used to calculate how good the prediction is, and its gradients are used to backpropagate throughout the network. It is important to define a good objective function for the network (LeCun and Huang, 2005). This process is standard for deep learning models. The parameters are updated with the formula:

$$\theta \leftarrow \theta - \lambda \nabla_\theta (L(E, E^*)) \qquad (2)$$

where θ is the network parameters, λ is the learning rate, and $\nabla_\theta(L(E, E^*))$ is the gradients of the loss between predicted and ground truth energies with respect to trainable parameters of the network.

4 Energy-Based Dialogue State Tracker

Based on the general principles of energy based modelling, we propose a deep learning energy-based architecture for dialogue state tracking. Given the approach outlined in the previous section, the model consists of three main components:

- **Feature network** is a function implemented as a deep learning network to transform dialogue input into an appropriate representation which can be fed to the energy function.
- **Energy function** is a function implemented as a feed-forward network that is trained to assign scalar energy values to any given configuration of input and output variables.
- **Loss function** is a function that provides an measurement of the quality of the network predictions.

In the following we provide details of these components as we specifically designed them for the DSTC2 dataset.

4.1 Feature Network

DSTC2 dialogue data consists of a number of calls (conversations) which in turn are built out of a sequence of turn pairs. Each turn pair consists of the user utterance itself, and a system response – referred to as the machine act.

User utterances are sequences of words (tokens); thus we use a bidirectional LSTM architecture (Hochreiter and Schmidhuber, 1997) to generate an initial representation of the whole word sequence in a turn (see Figure 2). This utterance LSTM is fed using a word embedding layer that is trained directly on our data; empirically we found this to provide us with better results than using a public pre-trained word embedding component.

Machine acts are provided in a semantic representation format, therefore we first parse these into vector representations following the approach outlined in the Word-based Dialogue state tracker (Henderson et al., 2014b). These machine act vectors are high-dimensional one-hot encodings; therefore we find it useful to feed these through an encoder to produce a reduced distributed representation (see Figure 3).

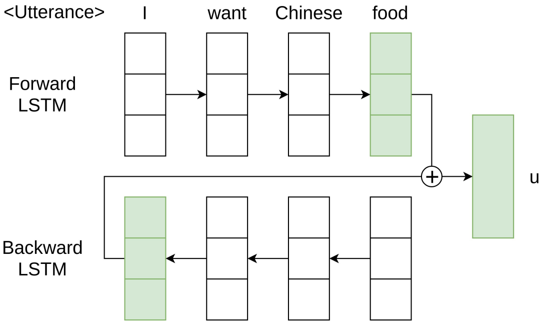

Figure 2: The bidirectional LSTM architecture to encode utterances. $\oplus$ denotes the concatenation operation.

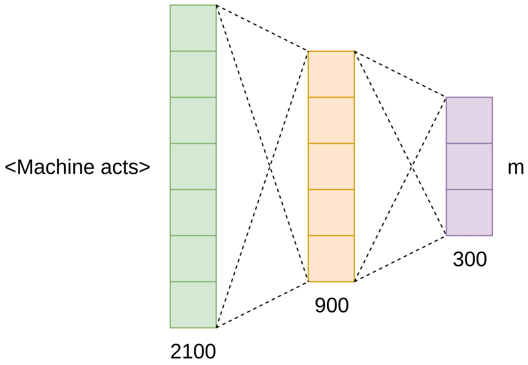

Figure 3: The encoder with two fully connected layers to reduce the dimensionality of machine act vectors.

We concatenate the encoded machine act vector with the output vector of the bi-directional utterance encoder to form a dialogue turn representation vector.

In order to handle dialogue input and dialogue history, it is necessary to use a second LSTM layer unrolling throughout individual turns to build up a complete representation of the dialogue (see Figure 4). Therefore, we feed the input vector produced for each turn into the second full-dialogue LSTM, and receive a fixed-size output vector – this is thus a representation of the whole dialogue up to the current turn. Hyper-parameters for the two LSTM layers plus the embeddings layers used to produce distributed representations of both user utterances and machine acts are presented later in Table 2.

While it is possible for us to feed the output of the second LSTM layer directly as input to an energy layer and perform training, this approach is sub-optimal. As noted by Belanger and McCallum

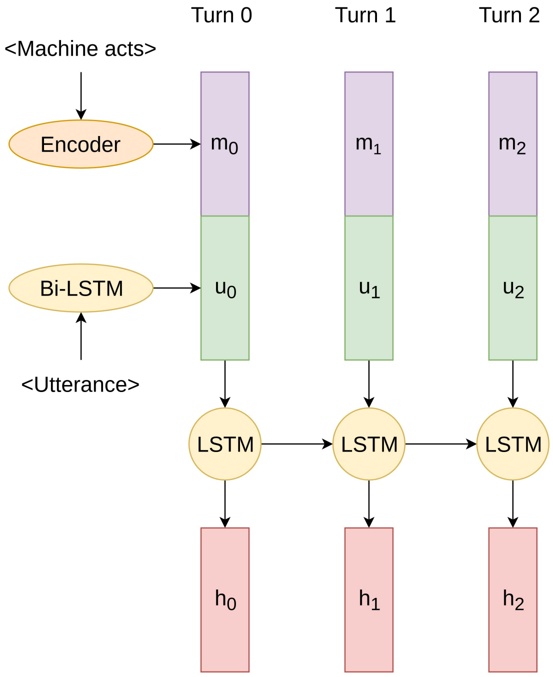

Figure 4: The deep LSTM architecture to transform dialogue input into fixed-size vector representations.

(2016), the feature network should ideally be pre-trained to improve the quality of features. Therefore we pre-train our feature network by plugging it into a multi-task style learning architecture for dialogue state tracking in the style of that proposed by Trinh et al. (2018). Specifically, to complete pre-training the outputs of the second LSTM are fed to a set of three softmax outputs that affect three independent multinomial targets. Optimisation with backpropagation is then used to train the network in the normal way. When used as input to the energy network, the final layer consisting of a set of three softmax operations are discarded and instead the LSTM outputs are taken to be the output of the feature network, i.e., $F(X)$.

The above approach has the advantage that the feature network's output vectors $F(X)$, i.e., the outputs of the turn based LSTM, are already well aligned to producing candidate target representations Y – although they are not actual candidate targets.

4.2 Energy Function

The energy function is implemented on top of the feature network to assign the scalar energy values to combinations of dialogue input and output variables. It should be noted though that the energy function in the literature is usually defined in terms of X and Y, but, for the sake of clarity, we will describe it in terms of Y and $F(X)$, our pre-trained feature representation.

We build our model based around that proposed for the Structured Prediction Energy Network (SPEN) model (Belanger and McCallum, 2016). In this approach the energy function is the summation of individual *Local energy* and *Global energy* terms:

$$\mathcal{E} = E_{local}(F(X), Y) + E_{global}(Y) \qquad (3)$$

Local energy is computed between input and output (label) variables.

$$E_{local}(F(X), Y) = \sum_{i=1}^{L} y_i W_i^\top F(X) \qquad (4)$$

where W_i is a vector for each label, and $y_i \in Y$ is the i^{th} label in the label set.

Global energy meanwhile captures the relationship between labels in the set of output variables independently of the input features. It is also called Label energy and is given below:

$$E_{global}(Y) = W_{g2}^\top \tanh(W_{g1}^\top Y) \qquad (5)$$

where all weights W, W_{g1}, and W_{g2} are parameters that are learned during the training process.

4.3 Loss Function

There are several options for designing the loss function for use in energy-based modelling. In our architecture, we use a loss function based on that proposed for the end-to-end SPEN model (Belanger et al., 2017). This is given as follows:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{T-t+1} L(y_t, y^*) \qquad (6)$$

where T is the number of iterations in the inference process, t is an iterative variable running through the inference loop, and $L(y_t, y^*)$ is the loss function between the predicted output and the target labels.

The motivation for this loss function is that it measures the quality of every generated prediction $L(y_t, y^*)$ in the inference loop, and encourages the Energy function to produce good quality prediction by including the coefficient for each iteration $\frac{1}{T-t+1}$.

Although the end prediction y_T is our desired output, it is not advised to only calculate loss value of this output. If doing so, the model can possibly generate the output only at the last inference iteration rather than moving smoothly towards the output in the loop.

Since we define the dialogue state tracking task as a multilabel classification task, we use the cross entropy loss for the formula $L(y_t, y^*)$.

5 Experiments

In this section we provide details of the dataset, hyper-parameter selection, and validation results. Test results are presented in the next section.

We train our models with the training set and use the development set to select the best trained parameters. Following this, we run our models with the test set and report those results.

For the food type, price range, and area slots, we merge all three labels into a single multi-label classification task for the sake of the energy-based calculations. In other words we sacrifice the domain constraint that one and only variable can be active individually for each of our slots and instead look for complete global configurations. This is necessary to allow a more elegant integration with the energy-based mechanisms we introduced in the previous sections. In practice our model still (mostly) learns that we need one and only one slot for each of the *food*, *price range*, and *area* related subspace of our target variable.

The model performance is evaluated and reported with the *accuracy* metric, which is one of the feature metrics for the DSTC2.

5.1 Model hyper-parameters

As indicated earlier, we developed a multi-task deep learning state tracker to pre-train the feature network which is subsequently supplied to the energy-based network. This network in practice also serves as a valid benchmark against which we can compare the results of our energy-based model.

This multi-task learning network consists of our feature network (section 4.1) leading into three classifiers for the three informable slots. These three classifiers are implemented with *softmax* output activation function as tracking each slot by itself is a multinomial classification task. We train all parameters of this system end-to-end with a cross entropy loss function and use the Adam optimizer.

The energy-based system is trained with the best set of pre-trained parameters from the multi-task learning-based system having reviewed its performance on the DSTC2 development set. As we combine the labels of informable slots, the task then becomes a multilabel classification task. Therefore we use a *sigmoid* activation function for the output of the energy-based system to produce predictions rather than using three softmax functions as used in the multi-task network above.

The detail of the selected hyper parameters are presented in the Table 2. All hyper parameters are chosen through a strict selection based on the experiments on DSTC2 training and development sets. We developed our energy-based model in TensorFlow (TF) 1.13 (Abadi et al., 2015). As is the case with the multi-task system, we apply the cross entropy loss function and the Adam optimizer (Kingma and Ba, 2015) to train the energy-based network.

Hyper parameter	Value
Feature network	
Machine acts encoded size	300
Encoder output activation	$\tanh$
Word embedding size	300
LSTM number of units	128
LSTM drop out	0.2
LSTM output activation	$\tanh$
Inference process	
Number of iterations	50
Initial learning rates	0.001
Non-linearity function	$\tanh$
Learning process	
Loss function	Cross entropy
Optimizer	Adam
Learning rate	0.001
Maximal global gradient norm	5.0

Table 2: Basic hyper parameters used in experiments constructing the energy-based dialogue state tracker.

5.2 Validation results

During the development phase we carry the evaluation of our multi-task learning-based and energy-based models against the DSTC2 development set in order to find the best set of parameters. We report on both a mean accuracy produced with Tensorflow directly from our data, and the Joint Goals accuracy produced by the toolset provided for the DSTC2 dataset (Henderson et al., 2013). We present the validation results in Table 3.

In the validation results we observe that ap-

Model	TF Acc.	DSTC2 Acc.
Multi-task	0.719	0.692
Energy-based	0.759	0.715
DSTC2 Baseline		0.623

Table 3: Model performances on the Joint Goals task of DSTC2 development set.

plying the energy network on top of deep learning feature network improves the accuracy on the main tracking task by a margin up to 4%. We also see that there is a big gap between raw accuracy during the training process and external DSTC2 joint goal accuracy results when running evaluation on the output track file. This can be explained by a number of factors, including our exclusion of one of the informable slots from the DST task, that brings the accuracy on the DSTC2 development set down by nearly 1%, and the fact that the raw accuracy metric is carried on mini-batches while the DSTC2 metric evaluates the output of the whole dataset. Despite the differences, it is clear that the overall indicative result indicates a strong improvement with the application of the energy network.

6 Results & Discussions

We selected the best fitting set of hyper-parameters and the highest accuracy checkpoint from validation for use on the test set. We report our results against the DSTC2 baseline and other state-of-the-art trackers (see Table 4). We choose reference dialogue state trackers that are related to our work in different aspects such as their investigation of variable dependencies or because the network architecture is similar to or inspired that which we use. The evaluation metric used on test results is the accuracy provided by the DSTC2 reference evaluation system since this is the same metric used by the published solutions.

Similar to the development set, the energy-based model outperforms the multi-task deep learning tracker by a large margin. The observed improvement can only be achieved due to the energy function and inference process of the energy-based learning approach. Our multi-task learning-based tracker is developed with a straight-forward recurrent neural networks (RNN) architecture. The multi-task model is trained to track all three DSTC2 informable slots at the same time, but it does not really tackle the relationships between them. On the other hand, the energy-based network includes the possible dependencies of these slots by using an energy function over all slot labels and pre-trained features.

Model	Accuracy
Hybrid Tracker	0.796
Word-based Tracker	0.768
EncDec Framework	0.730
MTL Model	0.728
CRF Tracker	0.601
Our work	
Energy-based Tracker	0.749
Multi-task Tracker	0.720
DSTC2 Baseline	0.719

Table 4: The performances of Dialogue State Trackers on the Joint Goals task of DSTC2 test set.

As mentioned above there exist Dialogue State Trackers that also tackle the relationships between variables such as EncDec Framework (Platek et al., 2016), MTL-based model (Trinh et al., 2018), and Conditional Random Field (CRF) tracker (Kim and Banchs, 2014). When comparing our energy-based model with those, we observe that our work achieves higher accuracy than those for the DSTC2 test set. Two out of three trackers, namely the MTL-based model and EncDec Framework, try to track Dialogue States within the incremental dialogue context, that limited their performances in general. Our work does not include the incrementality phenomenon. Kim and Banchs (2014) manually define input features in their work, that do not perform well. In our work we set up the model to learn these features automatically, and see improved results.

Among the state-of-the-art DSTC2 trackers, the Hybrid model (Vodolan et al., 2015, 2017) is the most similar in architecture to our work. Both approaches use a deep learning model as a feature network. The difference between their and our trackers lies in the algorithms applied on top of the feature network. For the hybrid tracker the authors apply a set of manual rule-based differentiable calculations to predict the dialogue states, while in our work we implement an energy network, that is also deep learning-based. The Word-based tracker (Henderson et al., 2014b) is a fully RNN-based model, that is notable for its high performance and the feature extraction technique. Vodolan et al. (2017) as well as our work adopts this technique to extract features from dialogue input.

6.1 Variable Associations Analysis

As observed above, the energy-based system performs better than the multi-task model in overall score of accuracy. However, the accuracy metric does not provide any extra information in terms of variable associations that the energy-based approach takes advantage of. Therefore, we performed further analysis on the results that our trackers produced for DSTC2 test set to compare our predictions to those of the DSTC2 baseline system. The analysis is conducted in a similar fashion to that presented in section 2.2, and is presented in Table 5.

	food-price	food-area	price-area
Testset	0.609	0.658	0.428
Our work			
Energy	0.577	0.659	0.428
MTL	0.523	0.687	0.447
Baseline	0.497	0.657	0.389

Table 5: Result analysis of variable dependencies on the DSTC2 test set. The analysis is reported using the ϕ coefficient values for each informable slot pair. In the table, the first block is variable dependencies in labels of the test set, while the second block is variable dependencies detected by our energy-based (*Energy*) and multi-task (*MTL*) trackers, and the last block is the result of the best DSTC2 baseline system.

The analysis result demonstrates that our energy-based system is capable of tackling the presence of variable dependencies in DSTC2 test set. The energy-based method reflects the relationships of two informable slot pairs, *food – area* and *price range – area*, and produces a very close relationship for the other pair, *food – price range*. On the other hand, the multi-task learning approach manages to capture some dependencies that is shown in the result with bigger margins for all variable pairs.

Overall both the deep learning-based methods outperform the best DSTC2 rule-based baseline system in comparing variable dependencies in the tracking process for at least two out of three informable slot pairs of the task.

7 Conclusion

In this paper we presented an energy-based approach to Dialogue State Tracking task that improves the overall performance of a basic deep learning-based model. Energy-based Learning is notably good at structured prediction that we argue applies to the DST task. The results of our work strengthen the hypothesis that dependencies between variables within the dialogue context have an impact on dialogue state tracking performance. To our knowledge this is the first attempt to apply energy-based learning in a dialogue processing task. Though our results do not in themselves improve on the state of the art, the difference relative to a multi-task deep learning model is significant enough to indicate that the method could lead to improvements on the state of the art if combined with the state of the art. Beyond that combination with hybrid state-of-the-art models, there is other room for improvement. Our current plans includes the investigation of multivariate dependencies in dialogue processing with a larger domain and cross domains. We also believe that it is good to conduct an extensive analysis on variable dependencies in data and performances of different architectures.

Acknowledgements

This research was supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

David Belanger and Andrew McCallum. 2016. Structured Prediction Energy Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48.

David Belanger, Bishan Yang, and Andrew McCallum. 2017. End-to-End Learning for Structured Prediction Energy Networks. In *Proceedings of the 34th International Conference on Machine Learning*.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrksic, Tsung-Hsien Wen, Inigo Casanueva, Lina Rojas-Barahona, and Milica Gasic. 2017. Subdomain Modelling for Dialogue Management with Hierarchical Reinforcement Learning. In *Proceedings of the SIGDIAL 2017 Conference*, pages 86–92.

Jason J. Corso, Maneesh Dewan, and Gregory D. Hager. 2004. Image Segmentation Through Energy Minimization Based Subspace Fusion. In *Proceedings of the 17th International Conference on Pattern Recognition ICPR 2004*, volume 2, pages 120–123. IEEE.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2013. *Dialog State Tracking Challenge 2 & 3*.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The Second Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2014 Conference*, pages 263–272.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-Based Dialog State Tracking with Recurrent Neural Networks. In *Proceedings of the SIGDIAL 2014 Conference*, pages 292–299.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Takaaki Hori, Hai Wang, Chiori Hori, Shinji Watanabe, Bret Harsham, Jonathan Le Roux, John R. Hershey, Yusuke Koji, Yi Jing, Zhaocheng Zhu, and Takeyuki Aikawa. 2016. Dialog State Tracking With Attention-Based Sequence-To-Sequence Learning. In *Proceedings of 2016 IEEE Workshop on Spoken Language Technology*, pages 552–558.

Seokhwan Kim and Rafael E. Banchs. 2014. Sequential Labeling for Tracking Dynamic Dialog States. In *Proceedings of the SIGDIAL 2014 Conference*, pages 332–336.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. 2006. A Tutorial on Energy-Based Learning. *Predicting Structured Data*.

Yann LeCun and Fu Jie Huang. 2005. Loss Functions for Discriminative Training of Energy-Based Models. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AIStats'05)*, pages 206 – 213.

Qiuxu Li and Jieyu Zhao. 2009. MRF Energy Minimization for Unsupervised Image Segmentation. In *Proceedings of the 5th International Conference on Natural Computation, ICNC 2009*, volume 2, pages 67–73. IEEE.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-End Task-Completion Neural Dialogue Systems. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pages 733–743. Asian Federation of Natural Language Processing.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1064–1074.

Jiquan Ngiam, Zhenghao Chen, Pang Wei Koh, and Andrew Y. Ng. 2011. Learning Deep Energy Models. In *Proceedings of the 28th International Conference on Machine Learning*.

Ondrej Platek, Petr Belohlavek, Vojtech Hudecek, and Filip Jurcicek. 2016. Recurrent Neural Networks for Dialogue State Tracking. In *Proceedings of CEUR Workshop, ITAT 2016 Conference*, volume 1649, pages 63–67.

Robert J. Ross and John Bateman. 2009. Daisie: Information State Dialogues for Situated Systems. In *Proceedings of International Conference on Text, Speech and Dialogue, TSD 2009*, pages 379–386.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 3776–3783.

Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient Actor-Critic Reinforcement Learning with Supervised Data for Dialogue Management. In *Proceedings of the SIGDIAL 2017 Conference*, pages 147–157.

Anh Duong Trinh, Robert J. Ross, and John D. Kelleher. 2018. A Multi-Task Approach to Incremental Dialogue State Tracking. In *Proceedings of The 22nd workshop on the Semantics and Pragmatics of Dialogue, SEMDIAL*, pages 132–145.

Miroslav Vodolan, Rudolf Kadlec, and Jan Kleindienst. 2015. Hybrid Dialog State Tracker. In *Proceedings of the Machine Learning for SLU & Interaction NIPS 2015 Workshop*.

Miroslav Vodolan, Rudolf Kadlec, and Jan Kleindienst. 2017. Hybrid Dialog State Tracker with ASR Features. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, volume 2, pages 205–210.

Atro Voutilainen. 1995. A syntax-based part-of-speech analyser. In *Proceedings of the 7th conference on European chapter of the Association for Computational Linguistics EACL '95*, pages 157–164.

Jason D. Williams. 2010. Incremental Partition Recombination For Efficient Tracking Of Multiple Dialog States. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5382–5385.

Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning. In *Proceedings of the SIGDIAL 2016 Conference*, pages 1–10.

A Appendices

A.1 Dataset Analysis

We conduct a small analysis on the DSTC2 dataset to reason why we would like to choose only three out of four informable slots to track. In the analysis we count how often the slots appear in labels with a value, i.e. not *none*, and how often those slots change their values during the conversations.

	Slot appearance (%)			
	Food	Price	Area	Name
Value is not None				
dstc2_train	75.06	61.70	72.16	0.37
dstc2_dev	72.70	62.48	70.11	0.86
dstc2_test	87.01	63.82	73.25	0.51
Value is changed				
dstc2_train	17.12	10.10	11.50	0.07
dstc2_dev	15.56	9.23	10.24	0.20
dstc2_test	16.13	9.42	10.50	0.09

Table 6: The analysis of Informable slot appearances in DSTC2 dataset. The numbers are reported in the percent format (%) over the number of turns in the dataset.

Among DSTC2 informable slots, the slot *Name* rarely appears. That means the datset does not provide enough samples for training Deep Learning models to classify this slot. In the result, this slot does not affect the Joint Goals tracking performance, as in the DSTC2 test set predicting $Name = none$ gives 99.5% accuracy.

A.2 Chi-square Test

Chi-square test is a significant test for association between two variables. The task and algorithm are presented as follow.

Task Given a contingency table (table of counts) of two variables A and B. Let $P(A_i)$ and $P(B_j)$ are probability of appearance in the population of the categories A_i and B_j. Test the relationship between these two variables (dependent or independent).

Step 1 Define hypotheses of the task.

H_0: The two variables are independent

$$P(A_i \cap B_j) = P(A_i)P(B_j) \tag{7}$$

H_1: The two variables are dependent

$$P(A_i \cap B_j) \neq P(A_i)P(B_j) \tag{8}$$

Step 2 Calculate expected frequency of $\{A_i, B_j\}$ based on the input

$$E_{ij} = P(A_i) * P(B_j) * N \tag{9}$$

where N is the population.

Step 3 Calculate the chi-square error

$$\mathcal{X}^2_{\mathcal{V}} = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{10}$$

where $\mathcal{V}$ is degree of freedom, O_{ij} and E_{ij} are observed and expected frequencies subsequently.

Step 4 We reject H_0 if the computed test statistics $\mathcal{X}^2_{\mathcal{V}}$ is high and the significance coefficient $p < 0.05$.

There exist several measurements of association strength between variables directly related to the chi-square test statistics. There measures are scaled between 0 and 1 indicating that 1 is the perfect relationship and 0 is no relationship between variables. We choose ϕ coefficient to report the level of dependencies between slots in DSTC2 data as in section 2.2.

$$\phi = \sqrt{\frac{\mathcal{X}^2}{N}} \tag{11}$$

where $\mathcal{X}^2$ is the chi-square statistic value, and N is the number of samples in dataset.