TOWARDS AUTOMATIC METHODS TO DETECT ERRORS IN TRANSCRIPTIONS OF SPEECH RECORDINGS

Jinyi Yang[1] *Lucas Ondel*[3] *Vimal Manohar*[1,2] *Hynek Hermansky*[1]

[1] Center for Language and Speech Processing, Johns Hopkins University
[2] Human Language Technology Center of Excellence, Johns Hopkins University
[3]Brno University of Technology, FIT, IT4I Centre of Excellence

ABSTRACT

This work explores different methods to detect errors in transcriptions of speech recordings. We artificially corrupt well transcribed speech transcriptions with three types of errors: substitution, insertion and deletion on TIMIT phonemic transcriptions and WSJ word transcriptions. First, we use Bayesian model selection method by comparing the log-likelihoods from alignment and phone recognizer, a final score is computed to make decision. In this method, we consider two models, Bayesian Hidden Markov Model (HMM) and a Variational Auto-Encoder (VAE) combined with a HMM. Alternately, we build a biased ASR system with language models trained on individual transcriptions, detection decision is based on Levenshtein distance (LD) between transcription and oracle path from decoded lattice. We evaluate the methods of detecting errors in corrupted TIMIT transcription, the best result (either using model selection with VAE model or biased ASR) achieves 7% equal error rate on the Detection Error Tradeoff (DET) curve; we also evaluate the methods of detecting errors in corrupted WSJ transcriptions, and the best result (using biased ASR) achieves 3% equal error rate.

Index Terms— Transcription error detection, model selection, HMM-GMM, Variational Auto-Encoder, detection error tradeoff

1. INTRODUCTION

The quality of an Automatic Speech Recognition (ASR) system greatly depends on the training data. With the increasing needs for large amount of transcriptions of speech recordings, it is expensive and time consuming to obtain manual transcriptions. Considering the trade-off between efficiency and accuracy, many imperfect transcriptions are generated either manually or automatically by machines, then used as training data. Errors in training transcriptions may have great effects on final tasks. Previous research in [1] has shown that transcription filtering technique can benefit the ASR performance. Therefore, it is necessary to find automatic methods to detect erroneous transcriptions of speech recordings. Moreover, such methods may have a wide range of applications including text-to-speech (TTS) synthesis and semi-supervised training.

An intuitive way to find errors in transcriptions would be to use a well trained ASR system to decode the speech, and find the mismatches between transcription and decoding results. However, even the state-of-the-art ASR system cannot achieve zero error rate.

Previous works in [2] [3] use the information in transcriptions by performing Viterbi alignment. In [3], mean and standard deviation of the log-likelihood from the Viterbi alignment best path are used to detect transcription errors. The problem of this method is that the final decision depends only on the alignment log-likelihoods, which highly depends on the performance of the acoustic model, however, the acoustic model is not always reliable. In this work, we present two different approaches that remedy this shortcoming.

The paper is organized as follows. In Section 2, we describe our approaches. In Section 3, we describe experiments on detecting errors in both phone and word transcriptions. In Section 4 we present our conclusions.

2. METHODS

We considered two different approaches to find erroneous transcriptions: the first one is the well known *Bayesian model selection*, and the second one is a heuristic relying upon the lattice generated from a "biased" ASR system with a specific language model.

2.1. Bayesian model selection

Model selection is the task to select a model between two candidates $\{\mathcal{M}_1, \mathcal{M}_2\}$ given data. The process is formally described as:

$$\mathcal{M}^* = \begin{cases} \mathcal{M}_1 & \text{if } \mathcal{B} > 1 \\ \mathcal{M}_2 & \text{if } \mathcal{B} \leq 1 \end{cases} \tag{1}$$

$$\mathcal{B} = \frac{\int p(X|\theta_1, \mathcal{M}_1) p(\theta_1|\mathcal{M}_1) d\theta_1}{\int p(X|\theta_2, \mathcal{M}_2) p(\theta_2|\mathcal{M}_2) d\theta_2} \tag{2}$$

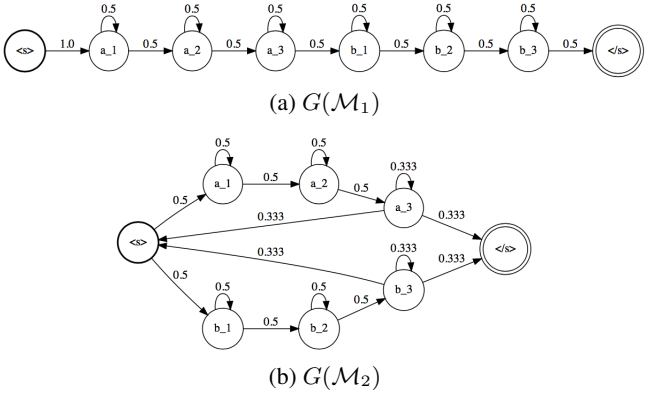
where $\mathcal{B}$ is the Bayes factor, θ_1 and θ_2 are models' parameters for $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively.

For our task, $\mathcal{M}_1$ and $\mathcal{M}_2$ are the same HMM based model but with different transition probabilities. For $\mathcal{M}_1$, we use a linear graph built from the transcription, as we perform forced alignment; for $\mathcal{M}_2$, we replace the linear graph with a *phone loop* structure, yielding to a phone recognizer with a uniform unigram phonotactic language model. Example of HMMs for both models are shown in Figure 1, denoted as $G(\mathcal{M}_1)$ and $G(\mathcal{M}_2)$.

Since the Bayes factor is intractable, as an approximation, we replaced the marginal probability in Equation 2 by the Variational Bayes (VB) lower bound (computed with the Viterbi approximation), which yields:

$$\ln \mathcal{B} \approx \mathcal{L}(\mathcal{M}_1) - \mathcal{L}(\mathcal{M}_2) \tag{3}$$

where $\mathcal{L}(\mathcal{M})$ is the VB lower bound. In our problem, when transcription matches well with the speech recording, the log Bayes factor will be close to zero (Viterbi paths from $\mathcal{M}_1$ and $\mathcal{M}_2$ will be similar) ; on the other hand, if the transcription does not match with speech recording, the log Bayes factor will be a large negative number. An illustration on this phenomenon is shown on a real case in



(a) $G(\mathcal{M}_1)$

(b) $G(\mathcal{M}_2)$

Fig. 1. Example of HMMs for $\mathcal{M}_1$ and $\mathcal{M}_2$ for an utterance with transcription "ab". Conventionally, each phone is represented with 3-states sub-HMM.

Figure 2 (a). As we see, the log Bayes factor is obviously negative when there are errors in transcription. In practice, we square the per-frame log-likelihood of the models to accentuate the difference between the correct and in correct transcriptions. The effect of such post-processing to obtain the final model selection score S is shown in Figure 2 (b). As we observe, correct and corrupted transcriptions can be well separated by final score Ss. Therefore the final score S can be used for making reliable detection decision. The whole process can be summarized as follows:

- Compute the Viterbi path $\hat{\pi}_1$ using $G(\mathcal{M}_1)$
- Compute the Viterbi path $\hat{\pi}_2$ using $G(\mathcal{M}_2)$
- Compute the final score for each utterance: $S = \sum_{n=1}^{N} S_n^2 = \sum_{n=1}^{N} (\ln p(x_n|\hat{\pi}_1) - \ln p(x_n|\hat{\pi}_2))^2$, where S_n is frame wise model selection score. Note that the final score is not normalized by utterance length, this is to make the short erroneous segments more distinct.

All these steps can be computed with any generative acoustic models, which makes this method easy to used in many application.

The decision about erroneous transcription detection in this method greatly depends on the chosen model. In this work we considered two alternatives: the Bayesian version of the traditional HMM-GMM [4] and the more recent VAE-HMM-GMM [5, 6]. To keep the notation uncluttered we denote these models respectively HMM and VAE-HMM hereafter.

2.2. Biased ASR system

Biased language models trained on imperfect transcriptions have been used in lightly supervised training of ASR systems [7]. A "biased" language model refers to building an individual language model for each utterance, with most phones(or words, depends on transcription type) only from its own transcription. It strongly biases the decoding results towards transcription. By decoding a test speech recording with such an biased ASR system, disagreements between acoustic model and its biased language model yield to errors in decoding result, hence we can identify the mismatches between the transcription and speech recording.

In our approach, we decode the utterance using the corresponding biased language model (and an pre-trained acoustic model) to generate a lattice instead of only the best path. From the lattice, we

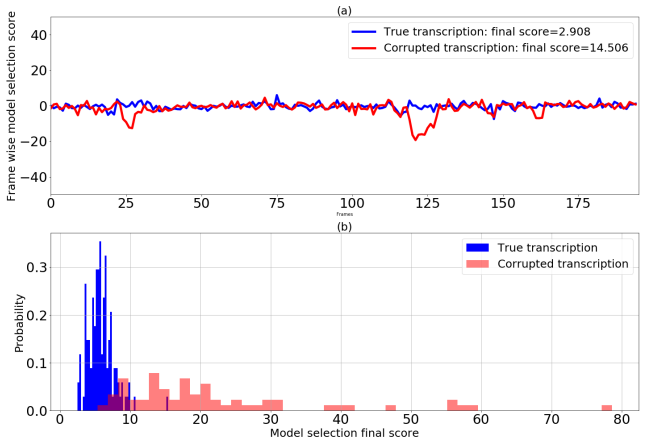


Fig. 2. Example of model selection score on true transcriptions and artificially corrupted transcriptions: (a) a single utterance, transcription has substitution errors between 25 to 28 frames and 120 to 126 frames; (b) 100 utterances, transcriptions have mixed three types of errors.

compute the *lattice oracle* unit error rate (unit can be either phone or word) using the algorithm in [1], which finds a path in the lattice that is the closest to the transcript in the Levenshtein distance, and this minimum distance is referred as the *lattice oracle* unit error. The error rate of the utterance is expected to be low if most of the transcription is contained in the lattice, which indicates that the transcription for the utterance is mostly correct.

3. EXPERIMENTS

3.1. Data preparation

Experiments to detect errors in phonetic transcription are done on TIMIT [8] corpus. TIMIT corpus contains continuous read speech in American English and phonetic transcriptions. Our training data contains 3696 utterances, approximately 3 hours, and test data contains 192 utterance, approximately 15 minutes. The TIMIT transcriptions are carefully examined to match with the speech, therefore we need to introduce artificial transcription errors in the test data as following: (1) Substitution. We use a phoneme confusion list described in [9], so that substitutions only occur between phonemes with high correlations; (2) Insertion. We count the frequency of all phonemes in training data, and choose the top 10 most frequent phonemes as insertion candidates; (3) Deletion. We randomly delete phonemes. We create four test sets with same speech recordings, but different errors in transcription, sentence error rate (SER) and phone

Test set name	Error type	SER (%)	PER (%)
test set true	N/A	0	0
test set 1	substitution	30	6
test set 2	insertion	30	6
test set 3	deletion	30	6
test set 4	mixed	30	6

Table 1. TIMIT test sets. "Test set true" is original TIMIT test set with true transcriptions; "mixed" is test set transcription with mixed three types of errors, each error has $\sim 2\%$ PER.

error rate (PER) of each set are shown in Table 1.

Experiments to detect of errors in word transcription are done on WSJ [10] corpus. WSJ corpus contains continuous read speech in English with word transcriptions. Our training set contains 37416 utterances, approximately 80 hours speech from WSJ's si-84 data subset, and test set contains 333 utterances, approximately 40 minutes speech from WSJ's evaluation 92 set. For this task we only create a test set with mixed errors in transcriptions, each error has $\sim$ 2% word error rate (WER), SER is 35%. Errors are introduced as following: (1) Substitution. We first compute the word frequency of true test transcriptions, then replace the top 30 most frequent words with candidates in the lexicon that have closest Levenshtein distance in pronunciation; (2) Insertion. We count the frequency of all words in training data, and choose the top 10 most frequent words as insertion candidates; (3) Deletion. We randomly delete words.

We use 39 dimensional Mel-frequency cepstral coefficients (MFCCs) + Δ + $\Delta\Delta$ coefficients as acoustic features.

3.2. Phoneme transcription error detection

3.2.1. Baseline

We use Kaldi [11] to perform phone recognition task. The acoustic model of baseline system is a HMM/DNN hybrid triphone system. The DNN is pre-trained in unsupervised fashion to initialize its parameters, then followed by a supervised fine-tuning. The DNN has five hidden layers with 1024 units per layer. The language model is a trigram phone language model. First we evaluate the performance of baseline system on "test set true", result is shown in Table 2. As we see, the PER is 19.1%, which is close to state-of-the-art according to [12]. Then we use the baseline system to obtain PER for each utterance in four corrupted test sets, if PER is above some threshold, we decide that this transcription is incorrect. Thresholds on PER are evenly spaced from 0 to 1 with an increment of 0.001 to draw the DET curve. This method is referred as "Baseline" in Section 3.2.4.

3.2.2. Model selection systems

We use two acoustic models to compute Viterbi alignment likelihoods and phoneme recognizer best path likelihoods: HMM and VAE-HMM.

For both models, the HMM is trained on 48 phonemes, with 1 silence phoneme and 47 non-silence phonemes. Silence phoneme is modeled by a 5 states left-to-right HMM, each state is modeled by a GMM with 10 components; each non-silence phoneme is modeled by a 3 states left-to-right HMM, each state is modeled by a GMM with 4 components.

Additionally for the VAE-HMM, the encoder and decoder have two linear layers with 512 units followed by Exponential Linear Unit (ELU) activation.

For training both models, the posterior distribution of the HMM's parameters θ are estimated using *Variational Bayes* (VB) training [4] (more precisely we used a stochastic variant described in [13] with batch size corresponding roughly to 400 utterances). The VAE-HMM is trained by combining the Stochastic VB algorithm together with the re-parameterized objective function of the standard VAE. See [14] for details about training VAE with probabilistic graphical model as prior over the latent space.

Again, we first evaluate the performance of both models on "test set true". Results are shown as in Table 2.Note that for both HMM and VAE-HMM monophone models, we simply use an uniform phonetic language model, yielding a purely acoustic phone recognizer. Therefore, the reported phone error rate is much higher compared to the baseline. Nonetheless, we will find that these models are competitive for the task of detecting errors in transcriptions.

The phone transcription error detection is processed as following: (1) compute the final score S for each utterance as described in Section 2.1; (2) when a final score is above some threshold, we decide that this is an incorrect transcription; (3) chose thresholds evenly spaced from 0 to an empirical value (which is larger than maximum of S, 100 in our experiment) with an increment of 0.01, draw the DET curve. These methods are referred as "Model selection: HMM" and "Model selection: VAE-HMM" in Section 3.2.4.

Besides, for comparison, we follow the previous work ([3]) and use the HMM and VAE-HMM monophone system to compute frame wise Viterbi alignment log-likelihoods for each utterance, variance of each log-likelihood is computed, if it is above some threshold, we decide that this transcription is incorrect. Thresholds on variances are evenly spaced from 0 to an empirical value (which is larger than maximum of variance, 300 in our experiment) with an increment of 0.01, and then used to draw the DET curve. These methods are referred as "Alignment llhs:HMM" and "Alignment llhs:VAE-HMM" in Section 3.2.4.

System	HMM monophone	VAE-HMM monophone	Baseline
PER(%)	37.77	37.61	19.10

Table 2. PER on TIMIT "test set true"

3.2.3. Biased ASR system

We use Kaldi to build the biased ASR system. The acoustic model is a triphone HMM model. The HMM topology is the same as used in model selection systems. The biased phone language model is built as follows: (1) for each utterance, we estimate a 4-gram unmodified Kneser-Ney interpolated language model from the transcript of this utterance; (2) this language model is then interpolated with an unigram language model estimated using counts of the top 10 most frequent phonemes in the whole training data set. Note that the unigram language model allows the decoding process to predict phoneme sequences that are not the same as the transcription, therefore the decoded lattice is more likely to include paths that are close to speech recording.

Similarly as used for baseline system, PER for each utterance is used as decision criterion, utterances with PER above some threshold are considered as incorrectly transcribed. Thresholds are evenly spaced from 0 to 1 with an increment of 0.001 to draw the DET curve. This method is referred as "Biased ASR" in Section 3.2.4.

3.2.4. Results

We draw DET curves and compute the Equal Error Rate (EER), namely the point where false alarm probability is equal to miss probability. The lower the EER is, the better the method is.

The EER of all methods on "test set 1", "test set 2" and "test set 3" are summarized in Table 3. Specially, for "test set 4", DET curves are shown in Figure 3.

We observed that on "test set 1", all methods have relatively poor performance. This is reasonable since the error type is substitution between easily confused phonemes. For any type of transcription errors, the "Biased ASR" always achieves the best performance, closely followed "Model selection: VAE-HMM". For test set with mixed type of errors, both "Biased ASR" and "Model selection: VAE-HMM" achieve the lowest EER (around 7%).

Test set name \ Method	Baseline	Alignment llhs: HMM	Alignment llhs: VAE-HMM	Model selection: HMM	Model selection: VAE-HMM	Biased ASR
test set 1	29.8	46.2	50.0	31.0	28.3	25.8
test set 2	23.8	27.2	23.4	6.0	3.0	2.0
test set 3	16.9	39.0	34.0	6.2	4.1	3.9

Table 3. EER (%) of different methods for phone transcription error detection.

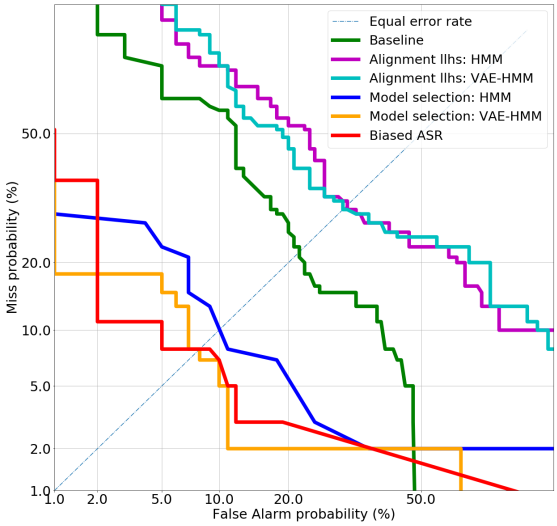


Fig. 3. DET curve of phone transcription errors detection on TIMIT "test set 4".

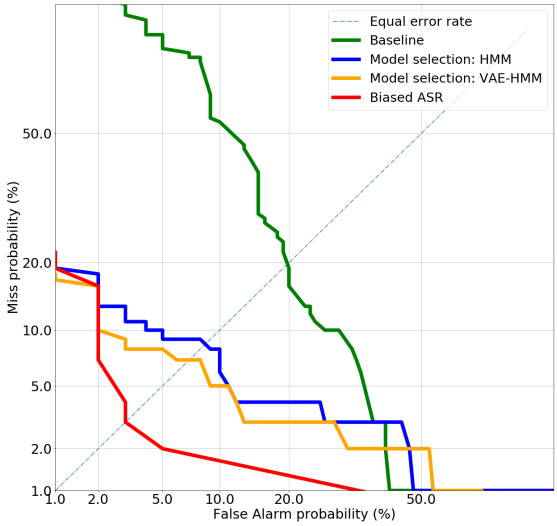


Fig. 4. DET curve of word transcription errors detection on corrupted WSJ evaluation 92 set.

3.3. Word transcription error detection

We use Kaldi to train the baseline word recognition system. The acoustic model is a HMM/DNN hybrid triphone system, the DNN has 4 layers with 1536 units in each layer. The language model is a trigram word language model. The WER of baseline system on correctly transcribed WSJ evaluation 92 set is 6.36%.

Both HMM and VAE-HMM models are trained on 42 phonemes, with 3 silence phonemes and 39 non-silence phonemes. HMM topologies, VAE-HMM neural network parameters, and training process are the same as described in Section 3.2.2. Since we do not use extra word language model, only PER can be evaluated on correctly transcribed WSJ evaluation 92 set. PER of HMM monophone system is 43.48%, PER of VAE-HMM monophone system is 39.46%.

The biased ASR is built following the same process as in Section 3.2.3, except that the 4-gram word language model is interpolated with an unigram word language model estimated using counts of the top 100 most frequent words in the whole training data set.

Results of word transcription error detection are shown in Figure 4. We compare the following methods: (1) use baseline system to decode test speech (referred as "Baseline"); (2) model selection using HMM and VAE-HMM models (referred as "Model selection: HMM" and "Model selection: VAE-HMM"); (3) biased ASR (referred as "Biased ASR"). In this case, the biased ASR achives best EER (around 3%). This result is to be expected as our model selection methods of the HMM and VAE-HMM is purely implemented on acoustic analysis and do not make use of word language model.

4. CONCLUSION

We introduced a Bayesian model selection approach to detect erroneous transcriptions of speech recordings. Despite relying solely on the acoustic information, this method achieves $\sim 7\%$ EER on phone and word transcription error detection. This method outperforms baselines relying on the output of a strong ASR system or the log-likelihood of the Viterbi alignment. Also as expected, we observed that language model is beneficial for word errors detection as used in the biased ASR. Future work will be extended in following aspects: refine the model selection method by incorporating language model to expect improvement over heuristically derived biased ASR approach; improve the granularity of the detection by finding accurate time boundaries of the errors; correct the detected errors.

5. ACKNOWLEDGEMENT

The work was supported by the gift funding from Beijing Magic Data Co., Ltd, and the Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

6. REFERENCES

[1] Vijayaditya Peddinti, Vimal Manohar, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Far-field ASR without parallel data," in *Proc. INTERSPEECH*, 2016.

[2] Michele Gubian, Barbara Schuppler, JJHC van Doremalen, EP Sanders, and LWJ Boves, "Novelty detection as a tool for detection of orthographic transcription errors," 2009.

[3] Swetha Tanamala, Jeena J Prakash, and Hema A Murthy, "A semi-automatic method for transcription error correction for indian language tts systems," in *Communications (NCC), 2017 Twenty-third National Conference on*. IEEE, 2017, pp. 1–6.

[4] Lucas Ondel, Luka Burget, and Jan ernock, "Variational inference for acoustic unit discovery," *Procedia Computer Science*, vol. 81, pp. 80 – 86, 2016, SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.

[5] Thomas Glarner, Patrick Hanebrink, Janek Ebbers, and Reinhold Haeb-Umbach, "Full bayesian hidden markov model variational autoencoder for acoustic unit discovery," *Proc. Interspeech 2018*, pp. 2688–2692, 2018.

[6] Lucas Ondel, Pierre Godard, Laurent Besacier, Elin Larsen, Mark Hasegawa-Johnson, Odette Scharenborg, Emmanuel Dupoux, Lukás Burget, François Yvon, and Sanjeev Khudanpur, "Bayesian models for unit discovery on a very low resource language," *CoRR*, vol. abs/1802.06053, 2018.

[7] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[8] John S Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[9] Carla Lopes and Fernando Perdigão, "Broad phonetic class definition driven by phone confusions," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 158, 2012.

[10] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[12] Josef Michalek and Jan Vanek, "A survey of recent dnn architectures on the timit phone recognition task," *arXiv preprint arXiv:1806.07974*, 2018.

[13] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303–1347, May 2013.

[14] Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta, "Composing graphical models with neural networks for structured representations and fast inference," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 2946–2954. Curran Associates, Inc., 2016.