# Exploring Energy-based Language Models with Different Architectures and Training Methods for Speech Recognition

*Hong Liu*[1,†], *Zhaobiao Lv*[2,†] , *Zhijian Ou*[*,1], *Wenbo Zhao*[2], *Qing Xiao*[2]

[1]Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, Beijing, China
[2]China Unicom (Guangdong) Industrial Internet Co., Ltd.

liuhong21@mails.tsinghua.edu.cn, lvzb7@chinaunicom.cn, ozj@tsinghua.edu.cn,
{zhaowb19, xiaoq17}@chinaunicom.cn

## Abstract

Energy-based language models (ELMs) parameterize an unnormalized distribution for natural sentences and are radically different from popular autoregressive language models (ALMs). As an important application, ELMs have been successfully used as a means for calculating sentence scores in speech recognition, but they all use less-modern CNN or LSTM networks. The recent progress in Transformer networks and large pretrained models such as BERT and GPT2 opens new possibility to further advancing ELMs. In this paper, we explore different architectures of energy functions and different training methods to investigate the capabilities of ELMs in rescoring for speech recognition, all using large pretrained models as backbones. Extensive experiments are conducted on two datasets, AISHELL-1 and WenetSpeech. The results show that the best ELM achieves competitive results with the finetuned GPT2 and performs significantly better than the finetuned BERT. Further analysis show that the ELM obtains better confidence estimate performance than the finetuned GPT2.

**Index Terms**: energy-based language model, rescoring, speech recognition

## 1. Introduction

Energy-based language models (ELMs), as a class of energy-based models (EBMs) [1, 2], parameterize an unnormalized distribution up to an unknown normalizing constant for natural sentences [3, 4, 5]. ELMs are radically different from popular autoregressive language models (ALMs), which are locally normalized. Unfortunately, local normalization in ALMs brings some drawbacks, e.g., ALMs are prone to exposure bias [6, 7] and label bias [8, 9]. ELMs potentially address these issues, as they do not require any local normalization. However, both exact computation of the normalizing constant and exact generation of samples from ELMs are generally intractable, which makes training especially difficult for ELMs.

In recent years, there are encouraging progresses in both theories and applications of ELMs. Applications of ELMs have covered computation of sentence likelihoods (up to a constant) [4, 5, 10, 11, 12, 13], text generation [14], language model pretraining [15], calibrated natural language understanding [16], and so on. As an important application, ELMs have been successfully used as a means for calculating sentence scores in automatic speech recognition (ASR). [4, 5] proposes trans-dimensional random field language model (TRF-LM) and applies to rescoring (i.e., reranking) of n-best lists for speech recognition. TRF-LMs outperform modified Kneser-Ney smoothing n-gram models [17] when both using n-gram features. Early ELMs are log-linear models [3, 4, 5]. Later, ELMs using neural network based energy functions have been developed [10, 11, 12], outperforming ALMs with similar model sizes, but they all use old-fashioned CNN or LSTM networks. The recent progress in Transformer networks [18] and large pretrained models such as BERT [19] and GPT2 [20] opens new possibility to further advancing ELMs. In this paper, we explore different architectures of energy functions and different training methods to investigate the potential capabilities of ELMs in rescoring for speech recognition.

The architectures of energy functions in ELMs can be very flexibly defined. In this work, we summarize and improve a suite of ELM architectures and name them SumTargetLogit, Hidden2Scalar, SumMaskedLogit and SumTokenLogit respectively. Model training of ELMs is challenging due to the intractable normalizing constant. We leave detailed discussions to the sections of Related Work and Methods.

Extensive experiments are conducted on two widely used Chinese speech recognition datasets, AISHELL-1 [21] and WenetSpeech [22]. We adopt large pretrained language models (PLMs) as the backbones of all energy models, noise models and proposal models in this work. We compare different combinations of architectures and training methods on these two datasets. The results show that the best ELM achieves competitive results with the finetuned GPT2 and performs significantly better than the finetuned BERT. The advantage of ELM is more obvious on the large-scale WenetSpeech. Further analysis show that the ELM obtains better confidence estimate performance than the finetuned GPT2.

## 2. Related work

**Architectures of ELMs**

One is generally free to choose the energy function, as long as it assigns a scalar energy to every sentence. TRF-LM [12] builds ELM in different dimensions according to sentence lengths, and directly sum the logit output from an ALM as energy, which corresponds to the SumTargetLogit architecture in this paper. Electric [15] is not strictly an ELM over sentences. It is in fact a cloze model, using contextualized encoder outputs to define conditional energies. Electric leverages the pseudo-log-likelihood (PLL) [23] to score the sentence, which inspires the SumMaskedLogit and SumTokenLogit architectures. In [16], three variants of energy functions are introduced. The first corresponds to the Hidden2Scalar architecture, and the latter two are based on a classification model, which is not relevant to the rescoring task in this paper. [14] proposes a residual energy based model for conditional text generation, where a residual

† Equal contribution. This work is supported by NSFC 61976122.

* Corresponding author: Zhijian Ou. The code is released at https://github.com/thu-spmi/CAT/blob/master/docs/energy-based_LM_training.md.

energy is defined on top of an ALM. In addition to ELMs, there have also existed energy-based end-to-end speech recognition models for ASR, for which we refer readers to [24, 25].

**Training methods for ELMs**
There are two mainstream training methods, maximum likelihood estimate (MLE) and noise contrastive estimate (NCE) [26]. In MLE, calculating gradients of the log likelihood usually resorts to Monte Carlo sampling methods. Two widely-used classes of sampling methods are importance sampling (IS) and Markov Chain Monte Carlo (MCMC) [27]. MCMC covers a range of specific algorithms, e.g., Metropolis independent sampling (MIS) is a special instance, where the proposed move is generated independent of the previous state. NCE, on the other hand, fitting unnormalized models by learning from distinguishing data samples and noise samples, where a noise distribution is required. Dynamic NCE (DNCE) [12] is an extension of NCE, which updates the noise distribution dynamically during training.

# 3. Methods

Let $x$ be a natural sentence (i.e., a token sequence). An energy-based language model (ELM) is defined as follows

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)} \tag{1}$$

where $E_\theta(x)$ denotes an energy function with parameter $\theta$, $Z(\theta) = \sum_x \exp(-E_\theta(x))$ is the normalizing constant and $p_\theta(x)$ is the probability of sentence $x$. The design of $E_\theta(x)$ and the optimization of $\theta$ are the focus of this work.

**TRF-LM**: [4] proposes trans-dimensional random field language model (TRF-LM), which builds energy models in different dimensions according to sentence lengths. Let $|x|$ be the length of sentence $x$, TRF-LM is defined as

$$p_\theta(x) = \pi_{|x|}\frac{\exp(-E_\theta(x))}{Z_{|x|}(\theta)} \tag{2}$$

where $Z_{|x|}(\theta)$ is the normalizing constant for length $|x|$. $\pi_{|x|}$ is the prior probability of length $|x|$, which is usually set as the empirical length probability, calculated from training data. The motivation of introducing length probabilities $\pi_{|x|}$ is that the empirical length probabilities can serve as a control device to improve sampling from multiple distributions over different lengths [4, 5]. To be differentiated from TRF-LM, the model in Eq. 1 is called globally-normalized ELM (**GN-ELM**).

## 3.1. Architectures of Energy Functions

The architectures of energy functions in ELMs can be very flexibly defined. In the following, we summarize and introduce some architectures for ELMs. Let $x = \{x_i\}_{i=1\dots|x|}$, where $x_i \in \{1, \cdots, V\}$ is the $i$-th token in $x$. $V$ denotes the size of token vocabulary. By abuse of notation, $x_i$ represents both the index of $x_i$ and the token itself.

**SumTargetLogit**: Similar to [12], we borrow the architecture from ALMs. Given history $x_{1:i-1}$, let the output logits to predict the next token be denoted by $f_\theta(x_{1:i-1})$, whose dimension is equal to $V$. The $k$-th logit is denoted by $f_\theta(x_{1:i-1})[k]$. Then, the energy is defined as

$$E_\theta(x) = -\sum_{i=1}^{|x|} f_\theta(x_{1:i-1})[x_i] \tag{3}$$

This energy function sums the logits corresponding to the target token (next token) at each position, hence it is named by SumTargetLogit. In contrast, the ALM applies local normalization (softmax) to the logits $f_\theta(x_{1:i-1})$ to obtain the conditional probability of $x_i$ given history $x_{1:i-1}$.

**Hidden2Scalar**: The energy of SumTargetLogit is defined in uni-directional order like in ALMs. More generally, like in [10, 14, 15, 16], we can use a bi-directional text encoder (e.g., BERT) to encode $x$ and we denote the encoder output (hidden vectors) by $\text{enc}_\theta(x)$. At position $i$, we have $\text{enc}_\theta(x)[i]$. Then, the energy is defined as

$$E_\theta(x) = -\text{Linear}\left(\sum_{i=1}^{|x|}\text{enc}_\theta(x)[i]\right) \tag{4}$$

where $\text{Linear}(\cdot)$ denotes a trainable linear layer whose output is a scalar.

**SumMaskedLogit**: For masked language model (MLM), e.g., BERT, pseudo-log-likelihood (PLL) is introduced for scroing sentences [23]. Inspired by this, we can define the energy function as follows:

$$E_\theta(x) = -\sum_{i=1}^{|x|} g_\theta(\text{MASK}(x,i))[i][x_i] \tag{5}$$

where $g_\theta$ denote the MLM, whose output, at each position, is the logits before softmax. $g_\theta(\text{MASK}(x,i))$ means masking the $i$-th token in $x$ and sending the masked sequence into the MLM for a forward pass. At position $i$, the logit corresponding to the masked token $x_i$ is denoted as $g_\theta(\text{MASK}(x,i))[i][x_i]$. Notably, this architecture is much time-consuming than others since it requires $|x|$ forward passes to calculate the energy of one sentence, therefore we do not experiment with this architecture.

**SumTokenLogit**: To overcome the deficiency of SumMaskedLogit, we propose a simplication, i.e., omitting the masking step and feeding $x$ directly to the MLM, so that the logits at all positions can be calculated in parallel. The energy is defined as:

$$E_\theta(x) = -\sum_{i=1}^{|x|} g_\theta(x)[i][x_i] \tag{6}$$

## 3.2. Training Methods

### *3.2.1. Noise Contrastive Estimate*

**Noise Contrastive Estimate (NCE)** [26] optimizes the ELM by learning from discrimination between data samples and noise samples. Let $q_\phi$ be the noise distribution with parameter $\phi$, the NCE objective is formulated as:

$$\mathcal{J}_{\text{NCE}}(\theta) = \mathop{\mathbb{E}}_{x\sim p_{\text{data}}} \log \frac{\hat{p_\theta}(x)}{\hat{p_\theta}(x)+\nu q_\phi(x)} + \nu \mathop{\mathbb{E}}_{x\sim q_\phi} \log \frac{\nu q_\phi(x)}{\hat{p_\theta}(x)+\nu q_\phi(x)}$$

where $\nu$ is the ratio between the noise prior and the data prior and $\hat{p_\theta} = \exp(-E_\theta(x))$ denotes the unnormalized probability. It is important that the noise distribution $q_\phi$ is close to the data distribution $p_{\text{data}}$ so that the binary classification task is sufficiently challenging for NCE to work [26].

**Dynamic NCE (DNCE):** DNCE [12] was proposed with two motivations. One is to push the noise distribution to be close to the data distribution; the other is to prevent the model overfitting to the empirical distribution when the training data cannot represent the oracle data distribution. DNCE modifies NCE from the above two aspects, and we only adopt the first modification in this paper, that is performing maximum likelihood optimization of $q_\phi$'s parameter over the training data

$$\mathcal{J}_{\text{DNCE}}(\theta,\phi) = \mathcal{J}_{\text{NCE}}(\theta) + \mathbb{E}_{x\sim p_{\text{data}}} \log q_\phi(x) \tag{7}$$

**Algorithm 1** Metropolis Independence Sampling in ELM.

**Input:** A target distribution $p_\theta$, a proposal distribution $q_\phi$, iteration number $T$.
Randomly initialize $x^{(0)}$;
**for** $t$ =1 to $T$ **do**
  Generate $x'$ from the proposal $q_\phi$;
  Accept $x^{(t)} = x'$ with probability $\min\{1, \frac{p_\theta(x')q_\phi(x^{(t-1)})}{p_\theta(x^{(t-1)})q_\phi(x')}\}$, otherwise set $x^{(t)} = x^{(t-1)}$;
**end for**
**Return:** $\{x^{(1)}, ..., x^{(T)}\}$

#### *3.2.2. Maximum Likelihood Estimate*

The gradient of log likelihood in MLE learning of ELMs can be derived as follows:

$$\frac{\partial \mathcal{J}_{\text{MLE}}(\theta)}{\partial \theta} = -\mathbb{E}_{x\sim p_{\text{data}}}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right] + \mathbb{E}_{x\sim p_\theta}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right] \quad (8)$$

The challenge is that the second expectation $E_{x\sim p_\theta}[\frac{\partial E_\theta(x)}{\partial \theta}]$ requires sampling from the unnormalized ELM $p_\theta$. Similar to [28], we compare two sampling approaches. Both methods need a proposal distribution $q_\phi$, which is implemented by an ALM in this paper. Note that the parameters of $q_\phi$ are also updated during training, similar to Eq. 7.

**Metropolis Independence Sampling (MIS)**: MIS is a special case of Metropolis-Hasting [27] and has been applied for ELM in [10]. Algorithm 1 shows the MIS details. In experiments, we run the Markov chain for $T$ steps, then use the samples $\{x^{(1)}, ..., x^{(T)}\}$ to approximate the expectation $E_{x\sim p_\theta}[\frac{\partial E_\theta(x)}{\partial \theta}]$ in Eq. 8 via Monte Carlo averaging.

**Importance Sampling (IS)**: different from the accept/reject sampling mode in MIS, IS [27] computes an importance weight $w(x') = \frac{p_\theta(x')}{q_\phi(x')}$ for each proposed sample $x'$. For $N$ proposed samples $x'_1, ..., x'_N$ from $q_\phi$, the second expectation in Eq. 8 is estimated as

$$\mathbb{E}_{x\sim p_\theta}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right] \approx \frac{\sum_{i=1}^N w(x'_i)\frac{\partial E_\theta(x'_i)}{\partial \theta}}{\sum_{i=1}^N w(x'_i)} \quad (9)$$

which theoretically is biased estimate. Note that we restart the chain after each parameter update in applying MIS, hence its gradient estimate is also biased. One research question to be addressed in this work is to compare MLE based on MIS and IS with NCE and DNCE for learning ELMs.

## 4. Experiments

### 4.1. Setup

**Datasets.** We AISHELL-1 [21] and WenetSpeech [22] in experiments. AISHELL-1 is a 178-hour mandarin speech dataset and WenetSpeech is a 1000+ hours multi-domain transcribed Mandarin speech dataset.

**ASR Model.** The main task of ELM is to rescore the n-best list output from the first-pass ASR decoding. We interpolate the score of ASR model, the score of language model and the sentence length to get the final score. In this work, the ASR n-best lists are obtained from a RNN-T [29] model, where the encoder is a Conformer [30] of 92M parameters. Note that the ASR model we use is rather competitive in terms of error rates of "no LM" on the two datasets in Table 1 and 2.

**Implementation Details.** We use Chinese GPT2 [20, 31] as the ALM $f_\theta$ in Eq. 3 and Chinese BERT [19] as the encoder and MLM in Eq. 4 and Eq. 6 respectively. Both pretrained models have 12 transformer layers with about 100M parameters. The noise distribution in NCE and the proposal distribution in MLE are both initialized from a finetuned GPT2, while the noise distribution in DNCE is initialized from a GPT2 without finetuning and is continuously optimized during training. We set the iteration number $T = 256$ in MIS sampling in Algorithm 1.

**Test Details.** On AISHELL-1 test set, we use the ASR model trained on AISHELL-1 and that trained on WenetSpeech to generate n-best lists respectively, i.e., in-domain test and cross-domain test, the results of which are denoted as $\text{CER}_1$ and $\text{CER}_2$ respectively in Table 1. As for WenetSpeech, we only use the ASR model trained on itself to generate n-best lists on the two test sets of WenetSpeech, TEST-NET and TEST-MEETING. The results are also denoted as $\text{CER}_1$ and $\text{CER}_2$ respectively in Table 2.

### 4.2. Results on AISHELL-1

Table 1 shows the Character Error Rate (CER) of different models on AISHELL-1 test set. The table is mainly divided into two parts. The first part is the rescoring results of non-energy models. Pretrained GPT2 and BERT represents that we rescore the n-best lists without any finetuning. Note that since BERT is not a traditional language model, we calculate the PLL mentioned above as the score of BERT. The second part is the results of ELMs trained with different methods and architectures on transcripts of AISHELL-1, which are explained in Sec 3.

It can be seen that among all the ELMs, the GN-ELM with Hidden2Scalar architecture and trained with DNCE achieves the lowest CERs, which are on par with the best results achieved by the finetuned GPT2, showing the competitiveness of ELM for scoring sentence. Besides, by observing and analyzing all the ELM experiments, we have the following conclusions:

**DNCE outperforms NCE.** We can see that the GN-ELMs trained with DNCE perform better than those trained with NCE under different architectures. This is not surprising since in DNCE, the binary classification challenge gradually increases with the optimization of noise model, which is more beneficial to optimizing ELM than a fixed noise model.

**GN-ELM and TRF-LM perform closely to each other.** When trained with DNCE, TRF-LM performs better than GN-ELM with SumTargetLogit, while it slightly underperforms GN-ELM with Hidden2Scalar and SumTokenLogit.

**MLE underperforms NCE/DNCE.** Most of the results of GN-ELMs trained with MLE-IS and MLE-MIS are worse than those trained with NCE/DNCE. Moreover, we find that the training process of MLE is quite unstable and not easy to converge when the hyper-parameters are not appropriately tuned. We attribute this to the difficulty of sampling in the high dimensional space of ELM (or say in other words, obtaining unbiased gradient estimates) within acceptable number of IS/MIS steps.

**Bi-directional architectures perform better.** The bi-directional architectures (Hidden2Scalar and SumTokenLogit) based on BERT are generally better than the unidirectional architecture (SumTargetLogit) except DNCE (TRF-LM).

### 4.3. Results on WenetSpeech

According to the conclusions above, we only conduct experiments of GN-ELM and TRF-LM trained with DNCE with different architectures on transcripts of WenetSpeech. In fact, we also conducted MLE experiments on WenetSpeech but the training did not converge with the loss tending to be negative infinite.

Table 1: *Rescoring results on AISHELL-1. $CER_1$ and $CER_2$ denote the Character Error Rate (CER) in in-domain test and cross-domain test respectively.*

| Method | Architecture | $CER_1$ | $CER_2$ |
|---|---|---|---|
| No LM | | 4.76 | 5.14 |
| 5-gram LM | | 4.67 | 4.40 |
| Pretrained GPT2 | | 3.22 | 3.66 |
| Pretrained BERT (PLL) | | 3.29 | 3.66 |
| Finetuned GPT2 | | 3.11 | 3.33 |
| Finetuned BERT (PLL) | | 3.12 | 3.47 |
| NCE (GN-ELM) | SumTargetLogit | 3.32 | 3.39 |
| | Hidden2Scalar | 3.20 | 3.36 |
| | SumTokenLogit | 3.27 | 3.43 |
| DNCE (GN-ELM) | SumTargetLogit | 3.25 | 3.40 |
| | Hidden2Scalar | 3.11 | 3.34 |
| | SumTokenLogit | 3.15 | 3.43 |
| DNCE (TRF-LM) | SumTargetLogit | 3.11 | 3.44 |
| | Hidden2Scalar | 3.13 | 3.39 |
| | SumTokenLogit | 3.21 | 3.47 |
| MLE-IS (GN-ELM) | SumTargetLogit | 3.42 | 3.61 |
| | Hidden2Scalar | 3.36 | 3.48 |
| | SumTokenLogit | 3.26 | 3.41 |
| MLE-MIS (GN-ELM) | SumTargetLogit | 3.35 | 3.59 |
| | Hidden2Scalar | 3.26 | 3.39 |
| | SumTokenLogit | 3.25 | 3.49 |

Table 2: *Rescoring results on WenetSpeech. $CER_1$ and $CER_2$ denote the CER in two test sets, TEST-NET and TEST-MEETING, respectively.*

| Method | Architecture | $CER_1$ | $CER_2$ |
|---|---|---|---|
| No LM | | 9.69 | 17.91 |
| Pretrained GPT2 | | 9.10 | 15.75 |
| Pretrained BERT (PLL) | | 9.07 | 15.69 |
| Finetuned GPT2 | | 8.82 | 15.52 |
| Finetuned BERT (PLL) | | 8.96 | 15.55 |
| DNCE (GN-ELM) | SumTargetLogit | 9.03 | 16.02 |
| | Hidden2Scalar | 8.98 | 15.69 |
| | SumTokenLogit | 8.81 | 15.47 |
| DNCE (TRF-LM) | SumTargetLogit | 8.97 | 15.77 |
| | Hidden2Scalar | 8.95 | 15.67 |
| | SumTokenLogit | 9.00 | 15.65 |

Table 2 shows the rescoring results on WenetSpeech. Different from the results on AISHELL-1, the GN-ELM with SumTokenLogit achieves the best results among all the ELMs and it even outperforms the finetuned GPT2. TRF-LMs do not achieve very good results on two test sets. Presumably, this is because the empirical length distribution obtained from the training data is not very applicable to the test set. Overall, the ELMs obtain the best performance upper on the large-scale WenetSpeech. This may reflect the benefit of ELMs in using bi-directional architecture and alleviating exposure bias and label bias.

### 4.4. Analysis and Discussion

**Significance Test.** To see whether the differences in Table 1 and Table 2 are significant, we conduct matched-pair significance test [32] for several pairs of experiments, whose $p$-values are shown in Table 3. If we set the significance level $\alpha = 0.05$, then all the experiment pairs with $p$-value less than 0.05 are considered be significantly different. Main observations are as follows: a) The best GN-ELM (both in Table 1 and Table 2) is significantly superior to the finetuned BERT in AISHELL-1 cross domain test and on the TEST-NET and TEST-MEETING of WenetSpeech, and it performs significantly better than the finetuned GPT2 on TEST-MEETING of WenetSpeech. Under other test situations, it achieves equally strong results as the finetuned GPT2 and BERT. b) The best TRF-LM is on par with the best GN-ELM in AISHELL-1 in-domain test but it underperforms the GN-ELM under other test situations significantly.

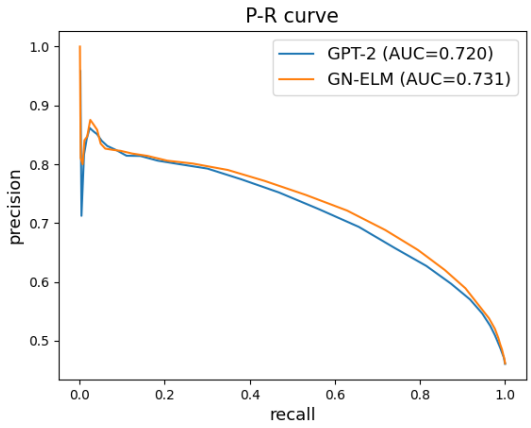


Figure 1: *The confidence estimate performace of the finetuned GPT2 and the best ELM on the TEST-NET of WenetSpeech.*

Table 3: *Matched pair test. The two p-values of each pair correspond to $CER_1$ and $CER_2$ in Table 1 and Table 2 respectively. A small p-value represents a more significant difference.*

| Dataset | Model Pairs | $p$-value | |
|---|---|---|---|
| AISHELL-1 | Finetuned GPT2<br>DNCE (GN-ELM) + Hidden2Scalar | 0.979 | 0.828 |
| AISHELL-1 | Finetuned BERT<br>DNCE (GN-ELM) + Hidden2Scalar | 0.821 | 1e-5 |
| AISHELL-1 | DNCE (TRF-LM) + SumTargetLogit<br>DNCE (GN-ELM) + Hidden2Scalar | 0.939 | 0.002 |
| WenetSpeech | Finetuned GPT2<br>DNCE (GN-ELM) + SumTokenLogit | 0.577 | 0.015 |
| WenetSpeech | Finetuned BERT<br>DNCE (GN-ELM) + SumTokenLogit | 1e-7 | 0.008 |
| WenetSpeech | DNCE (TRF-LM) + Hidden2Scalar<br>DNCE (GN-ELM) + SumTokenLogit | 1e-7 | 1e-7 |

**Confidence Estimate Performance.** The advantages of ELM also include confidence calibration [16, 33]. Following [34], we plot the precision-recall curve by changing the confidence threshold and calculate the AUC. The results are shown in Figure 1. The best GN-ELM on WenetSpeech achieves a higher AUC, which illustrates that the GN-ELM has a better confidence estimate performance than the finetuned GPT2.

## 5. Conclusions

In this paper, we explore energy-based language models with different architectures and training methods for rescoring in ASR. We summarize and improve several architectures and examine four different training methods, all using large pretrained models as backbones. Experiments are conducted on two widely-used datasets and the results show that the best ELM can achieve competitive results with the finetuned GPT2 and significantly better results than the finetuned BERT. We hope these new findings would be helpful for future work to further explore ELMs.

# 6. References


[1] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[2] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.

[3] R. Rosenfeld, S. F. Chen, and X. Zhu, "Whole-sentence exponential language models: a vehicle for linguistic-statistical integration," *Computer Speech & Language*, vol. 15, pp. 55–73, 2001.

[4] B. Wang, Z. Ou, and Z. Tan, "Trans-dimensional random fields for language modeling," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 785–794.

[5] B. Wang, Z. Ou, and Z. Tan, "Learning trans-dimensional random fields with applications to language modeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 876–890, 2017.

[6] S. Wiseman and A. M. Rush, "Sequence-to-sequence learning as beam-search optimization," in *EMNLP*, 2016.

[7] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *4th International Conference on Learning Representations (ICLR)*, 2016.

[8] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International conference on Machine learning (ICML)*, 2001.

[9] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2442–2452.

[10] B. Wang and Z. Ou, "Language modeling with neural trans-dimensional random fields," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 294–300.

[11] B. Wang and Z. Ou, "Learning neural trans-dimensional random field language models with noise-contrastive estimation," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[12] B. Wang and Z. Ou, "Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 70–76.

[13] S. Gao, Z. Ou, W. Yang, and H. Xu, "Integrating discrete and neural features via mixed-feature trans-dimensional random field language models," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[14] Y. Deng, A. Bakhtin, M. Ott, A. Szlam, and M. Ranzato, "Residual energy-based models for text generation," *arXiv preprint arXiv:2004.11714*, 2020.

[15] K. Clark, M.-T. Luong, Q. Le, and C. D. Manning, "Pre-training transformers as energy-based cloze models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 285–294.

[16] T. He, B. McCann, C. Xiong, and E. Hosseini-Asl, "Joint energy-based model training for better calibrated natural language understanding models," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 1754–1761.

[17] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[20] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[21] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, 2017, pp. 1–5.

[22] B. Zhang, H. Lv, P. Guo, Q. Shao, C. Yang, L. Xie, X. Xu, H. Bu, X. Chen, C. Zeng, D. Wu, and Z. Peng, "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022, pp. 6182–6186.

[23] A. Wang and K. Cho, "BERT has a mouth, and it must speak: BERT as a Markov random field language model," in *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 2019, pp. 30–36.

[24] H. Xiang and Z. Ou, "Crf-based single-stage acoustic modeling with ctc topology," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[25] E. Variani, K. Wu, M. D. Riley, D. Rybach, M. Shannon, and C. Allauzen, "Global normalization for streaming speech recognition in a modular framework," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4257–4269, 2022.

[26] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 297–304.

[27] J. S. Liu, *Monte Carlo strategies in scientific computing*. Springer, 2001.

[28] T. Parshakova, J.-M. Andreoli, and M. Dymetman, "Global autoregressive models for data-efficient sequence learning," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019, pp. 900–909.

[29] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[30] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[31] Z. Zhao, H. Chen, J. Zhang, X. Zhao, T. Liu, W. Lu, X. Chen, H. Deng, Q. Ju, and X. Du, "Uer: An open-source toolkit for pre-training models," *EMNLP-IJCNLP 2019*, p. 241, 2019.

[32] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1989, pp. 532–535.

[33] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," *arXiv preprint arXiv:1912.03263*, 2019.

[34] Q. Li, Y. Zhang, D. Qiu, Y. He, L. Cao, and P. C. Woodland, "Improving confidence estimation on out-of-domain data for end-to-end speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022, pp. 6537–6541.