INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Sentence Embedder Guided Utterance Encoder (SEGUE) for Spoken Language Understanding

Yi Xuan Tan, Navonil Majumder, Soujanya Poria

Singapore University of Technology and Design, Singapore
yixuan_tan@sutd.edu.sg, navonil_majumder@sutd.edu.sg, sporia@sutd.edu.sg

Abstract

The pre-trained speech encoder wav2vec 2.0 performs very well on various spoken language understanding (SLU) tasks. However, on many tasks, it trails behind text encoders with textual input. To improve the understanding capability of SLU encoders, various studies have used knowledge distillation to transfer knowledge from natural language understanding (NLU) encoders. We use a very simple method of distilling from a textual sentence embedder directly into wav2vec 2.0 as pre-training, utilizing paired audio-text datasets. We observed that this method is indeed capable of improving SLU task performance in fine-tuned settings, as well as full-data and few-shot transfer on a frozen encoder. However, the model performs worse on certain tasks highlighting the strengths and weaknesses of our approach.

Index Terms: spoken language understanding, knowledge distillation, pre-training

1. Introduction

Spoken language understanding (SLU) tasks [1] have greatly benefited from modern transformer-based speech encoders, such as wav2vec 2.0 [2], to the point that end-to-end models can now replace cascaded models, doing away with the ASR component. However, these models still trail behind equivalent tasks in the textual modality [3]. This could be interpreted as that textual models still contain much knowledge that speech models lack.

One way to transfer knowledge from one model to another is through knowledge distillation (KD) [4]. KD was originally used to transfer knowledge from larger models to smaller ones. However, with the assumption that parallel speech and text inputs are roughly equivalent, one could perform cross-modal KD from textual models to speech models. This idea has seen success in previous studies [5, 6, 7].

In this paper, we perform KD directly from a sentence embedder to a wav2vec 2.0 speech encoder. This produces a pre-trained sentence-embedder-guided utterance encoder (SEGUE), which can be used for sequence-level SLU tasks. We conduct experiments on SLU tasks—sentiment and emotion detection on speech modality, fluent speech commands (FSC), and automatic speech recognition (ASR). Our results demonstrate that SEGUE is capable of improving performance over vanilla wav2vec 2.0, to varying degrees. We show results for fine-tuned settings, as well as full-data and few-shot transfer with a frozen encoder. However, we also observe that SEGUE performs worse on two of said tasks. Our code is available on GitHub[1].

[1]https://github.com/declare-lab/segue

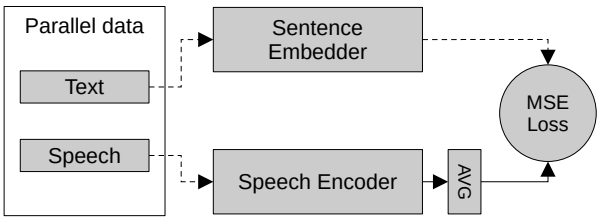


Figure 1: *Diagram of* SEGUE *pre-training. Solid arrows indicate gradient flow in the backward pass, and dashed arrows indicate the lack thereof.*

2. Related Work

Our approach aims to produce an utterance embedding, which was inspired by sentence encoding problems in the literature. In particular, we base the overall concept on Sentence-BERT [8]. We adopt the idea of knowledge distillation (KD) proposed by Hinton et al., and textual-to-spoken cross-modal KD has previously been explored by the following works. Cho et al. [5] perform KD directly with the downstream task, whereas we attempt to perform KD as pre-training for different downstream tasks to reuse. Denisov et al. [6] perform KD in pre-training as we do, but they construct an utterance encoder by initializing from a trained ASR model's backbone connected to a trained NLU backbone. In contrast, we attempt to distill knowledge directly into a wav2vec 2.0 encoder without ASR training and without a trained NLU module on top. Kim et al. [7] use a more complex architecture and perform KD in both pre-training and fine-tuning stages.

3. Method

Our method is based on the simple idea of distilling knowledge from a sentence embedder T into a speech encoder S, with the assumption that the textual input t and speech input s are roughly equivalent in meaning, as shown in Figure 1. To produce a single fixed-length embedding vector given a speech input s of length l, the sequence of speech encoder output vectors $[S_1(s), S_2(s), \ldots, S_l(s)]$ is average-pooled into a single embedding vector $S(s)$, similar to Sentence-BERT [8]:

$$S(s) = \frac{1}{l}\sum_{i}^{l} S_i(s) \tag{1}$$

A mean squared error (MSE) loss is then used to align the outputs of the speech student model to that of the textual teacher model.

$$L(s,t) = \|S(s) - T(t)\|_2^2 \tag{2}$$

10.21437/Interspeech.2023-1392

During pre-training, the loss $L(s, t)$ is computed, and its gradient w.r.t. the parameters of speech encoder S is computed to train said encoder, while sentence embedder T remains frozen.

We used the all-mpnet-v2-base checkpoint provided by the sentence-transformers package [8] as the sentence embedder. The speech encoder is a pre-trained wav2vec 2.0 base encoder as released by Baevski et al. [2]. Hence, both models produce a 768-dimensional embedding vector. Our model has 95 million parameters, equal to that of the wav2vec 2.0 base model.

This method requires text and speech of approximately equivalent meaning, so we used the 960 hours LibriSpeech dataset [9] as a source of parallel text and speech data.

One potential weakness of this method is that this may not capture the rich paralinguistic features in speech such as prosody, which are typically thought to be important for the semantic content of utterances. However, there may still be enough information in the training data for the model to learn to make use of these features to some extent.

3.1. Setting details

For pre-training, we used a linear LR schedule with warmup over the first 5000 steps, and a peak learning rate of 3e-5. The AdamW optimizer [10] was used with beta parameters 0.9 and 0.999, and 0 weight decay. We trained for 10 epochs on the LibriSpeech dataset. We used two A6000 GPUs with a per-device batch size of 8, totaling 16. The training took approximately 28 hours. We saved a checkpoint every 5000 steps, and average the parameters of the last 10 checkpoints to produce the final model.

3.2. Contrastive pre-training

Inspired by CLIP [11], our initial idea was to use the InfoNCE loss [12] to contrastively train a speech encoder and a text encoder to share the same embedding space. The task was set up such that the model has to predict, within a batch, which pairs of textual sentences and utterances correspond to each other, i.e. positive pairs. We monitored the loss, as well as the mean positive-pair similarity:

$$S_+(U, S) = \frac{1}{|U|} \sum_{i=1}^{|U|} S(u_i, s_j) \tag{3}$$

where $u \in U$ is a batch of utterances, $s \in S$ is a batch of textual sentences s.t. u_i and j_i form positive pairs, and $S(u_i, s_j)$ is the cosine similarity between the ith utterance embedding and the jth sentence embedding.

We found that this quickly causes catastrophic forgetting in the text encoder, and the speech encoder was not able to learn effectively from the setting. Therefore, we froze the text encoder, at which point it resembled a KD setting, but using InfoNCE rather than MSE loss, along with a learnable temperature for use with InfoNCE. We found that $S_+(U, S)$ correlated with good downstream task performance, but the loss did not. However, when we explored using KD with an MSE loss, it matched the best contrastive models that we had, and checkpoint selection with loss in KD was more consistent than with $S_+(U, S)$ in InfoNCE. Hence, we decided to use KD instead of contrastive training.

4. Experiments

We perform pre-training of SEGUE using the above method, and then evaluate the model on several downstream tasks, in fine-tuned settings (i.e. with a tunable encoder), as well as full-data and few-shot transfer settings with a frozen encoder (henceforth referred to simply as "full-data transfer" and "few-shot transfer"). We compare the results with the vanilla wav2vec 2.0 baseline. These tasks were done with the addition of a single linear layer on top of the encoder. We use the same AdamW setting as in pre-training, and a linear LR schedule with warmup unless otherwise stated. As both models were trained on 16 kHz mono audio, any audio input not in that format was converted into said format before being fed into the models.

4.1. Downstream tasks

We evaluate on sentiment regression with MOSEI [13], sentiment and emotion classification with MELD [14], intent classification with the MInDS-14 [15] en-US subset, and intent classification and slot-filling with Fluent Speech Commands (FSC) [16]. Additionally, we evaluated on ASR with FLEURS [17] as it may help highlight the weaknesses of SEGUE.

For MInDS-14, we randomly split the data into train-development-test subsets in a 60:20:20 ratio.

For multimodal tasks, i.e. MELD and MOSEI, we used only the speech modality, so we expected that the results would be far from state-of-the-art where the textual modality and potentially visual modality are used as well.

It should be noted that our copy of MOSEI has some missing videos, and due to the raw dataset no longer being publicly available, our results may not be fully comparable with results from other works.

5. Results

5.1. MOSEI

For the MOSEI sentiment task, the model was trained with a regression task, and rounding was used to retrofit the task as a classification task during evaluation. The fine-tuning and full-data transfer results are reported in Table 1. For full-data and few-shot transfer, we use an analytic solver for ridge regression with $\alpha = 100$.

For fine-tuning, we trained SEGUE for 3 epochs with 30% warmup steps, but we found that vanilla wav2vec 2.0 needed more epochs to converge so the baseline was trained for 5 epochs with 20% warmup steps. We trained with a peak LR of 3e-5 and a batch size of 8. We conducted three runs and report the mean and standard deviation of the metrics. We tried averaging the last 10 checkpoints, which we found tend to boosted overall performance but worsen MAE. Hence, we report both best-checkpoint and averaged-checkpoint results.

For both fine-tuning and full-data transfer, SEGUE improves performance across all metrics.

For few-shot transfer, we trained with k samples per class, under the seven-class setup, obtained by rounding the labels to the nearest integers. Due to the size of the data table, we instead report the results as plots – Figure 2a, Figure 2b, and Figure 2c. In terms of MAE and correlation, SEGUE learns faster than vanilla wav2vec 2.0. In terms of F1 score, SEGUE has a head start and maintains approximately the same lead as k grows.

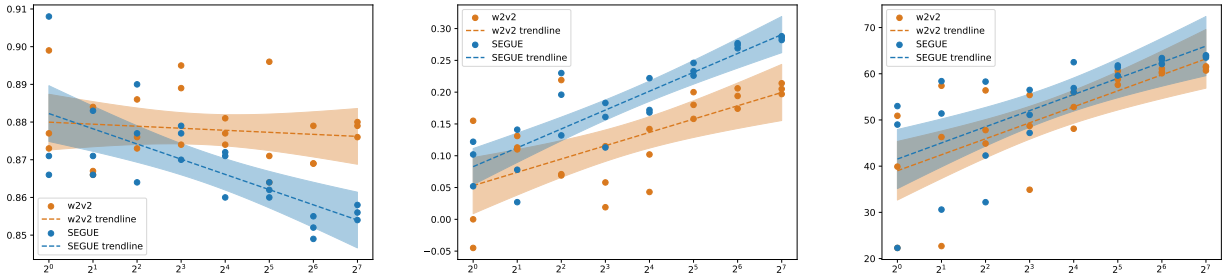
5.2. MELD

The results for fine-tuning and full-data transfer for MELD are shown in Table 2. SEGUE outperforms the baseline in both settings.

For fine-tuning, we trained for 5 epochs with 20% warmup steps and a batch size of 8. The wav2vec 2.0 baseline could not

Table 1: *Results of MOSEI fine-tuning. The number after ± indicates standard deviation.*

Model	MAE	Corr	F1	Acc2	Acc7
Fine-tuning					
w2v 2.0	.799 ± .006	.458 ± .016	72.9 ± 0.4	73.4 ± 0.2	37.5 ± 0.3
(averaged)	.799 ± .002	.460 ± .009	72.7 ± 0.4	73.0 ± 0.4	39.2 ± 0.9
SEGUE	**.781 ± .006**	**.473 ± .015**	73.1 ± 0.8	73.5 ± 0.3	**40.5 ± 0.5**
(averaged)	.797 ± .008	**.473 ± .006**	**73.5 ± 0.5**	**73.6 ± 0.5**	40.1 ± 1.5
Full-data transfer					
w2v 2.0	.855	.246	63.3	63.5	39.8
SEGUE	**.836**	**.326**	**65.5**	**65.4**	**40.5**



(a) *mean absolute error (MAE)* (b) *Pearson correlation coefficient* (c) *weighted F1 score*

Figure 2: *Plot of MOSEI k-shot per class performance against k.*

Table 2: *Results of MELD fine-tuning and full-data transfer. The number after ± indicates standard deviation.*

Model	Sentiment F1	Emotion F1
Fine-tuning (LR = 3e-6)		
w2v 2.0	46.8	37.2
(averaged)	47.3	39.3
SEGUE	**53.2**	41.0
(averaged)	**53.2**	**41.1**
Fine-tuning (LR = 3e-5)		
SEGUE	53.3	42.3
(averaged)	**54.1**	**47.2**
Full-data transfer		
w2v 2.0	45.0 ± 0.7	34.3 ± 1.2
SEGUE	**45.8 ± 0.1**	**35.7 ± 0.3**

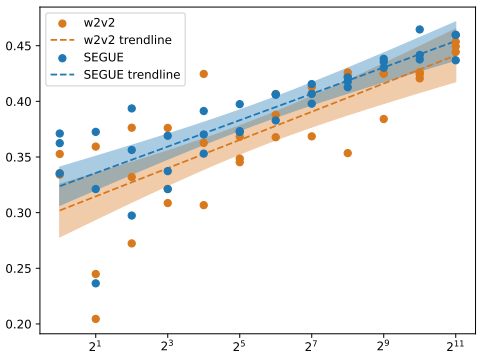


Figure 3: *Plot of MELD k-shot per class sentiment F1 score against k.*

converge with a learning rate of 3e-5 which we used for SEGUE, so we fine-tuned it at 3e-6 instead, and evaluated SEGUE in that setting as well. We also average the last 10 checkpoints and report the performance.

For full-data transfer, we trained for 20 epochs at a peak LR of 1e-3, with 10% warmup steps, and a batch size of 8. Due to the relatively close results and the training being somewhat noisy, we once again run thrice and report the average and standard deviation.

For few-shot transfer, we trained for 5 epochs at a peak LR of 1e-2 with 10% warmup steps. The results are once again presented as plots, in Figure 3 and Figure 4. The plots show that SEGUE has a head start, and mostly maintains that advantage as k grows.

5.3. Fluent Speech Commands

As with other fine-tuning settings where the results are close, we ran each experiment thrice. For fine-tuning, we trained for 5 epochs at a peak LR of 3e-5, with 10% warmup steps and a batch size of 8. For full-data transfer, we trained for 60 epochs at a peak LR of 1e-2, with 10% warmup steps and a batch size of 8.

As can be seen in Table 3, wav2vec 2.0 achieves approximately the same accuracy as SEGUE on fine-tuning. More surprisingly, wav2vec 2.0 is able to achieve 94.7% accuracy on full-data transfer even on a frozen encoder. In doing so it also outperforms SEGUE which only managed to reach 90.6% accuracy. We hypothesize that this is because the task can be performed

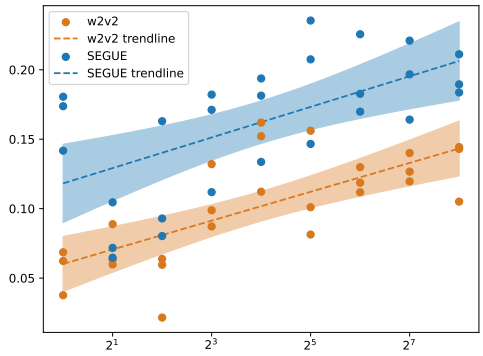


Figure 4: *Plot of MELD k-shot per class emotion F1 score against k.*

Table 3: *Results of FSC fine-tuning and full-data transfer. The number after ± indicates standard deviation.*

Model	Exact match accuracy
Fine-tuning	
w2v 2.0	**99.6 ± 0.0**
SEGUE	**99.6 ± 0.1**
Full-data transfer	
w2v 2.0	**94.7**
SEGUE	90.6

well with less understanding capability, relying more on word detection capability. The relative ease of this task as shown by the high accuracy scores may also point toward this hypothesis. Since SEGUE is trained against semantic embeddings alone, it is likely that it has lost some of the word detection capability of the original wav2vec 2.0 encoder, hence performing worse with a frozen backbone. However, it may not have completely forgotten the lost capability, hence it was still able to recover to the same performance as wav2vec 2.0 when fine-tuned.

Note that the above hypothesis may not apply to MInDS-14 despite it also being an intent classification task, as it contains utterances that are more free-form, unlike FSC's mostly fixed sentence structure, and thus may benefit more from deeper reasoning.

5.4. MInDS-14

The fine-tuning and full-data transfer results for MInDS-14 are shown in Table 4, and the few-shot transfer results in Figure 5.

For fine-tuning, we trained SEGUE for 40 epochs at a peak LR of 3e-5, with 10% warmup steps and a batch size of 16. The baseline once again needed longer to converge, so we trained for 60 epochs instead. We observed that the baseline could not stably converge, so we conducted 6 runs and broke down the results. On the other hand, SEGUE managed to stabilize the training, and improved the converged performance as well.

For both full-data and few-shot transfer, SEGUE drastically improves over the baseline.

5.5. FLEURS ASR

For FLEURS ASR, wav2vec 2.0 achieved a word error rate of 18.4, while SEGUE achieved 23.1. This is as expected, as SEGUE

Table 4: *Results of MInDS-14 fine-tuning and full-data transfer. The number after ± indicates standard deviation.*

Model	# runs	Accuracy
Fine-tuning		
w2v2 (all runs)	6	46.5 ± 47.0
w2v2 (failed runs)	3	3.5 ± 0.0
w2v2 (converged runs)	3	89.4 ± 2.3
SEGUE	3	**97.6 ± 0.5**
Full-data transfer		
w2v 2.0	–	54.0
SEGUE	–	**77.9**

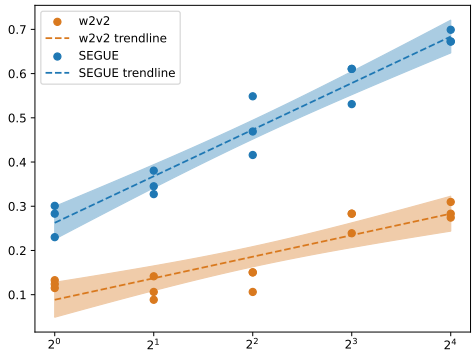


Figure 5: *Plot of MInDS-14 k-shot per class accuracy against k.*

was trained as a sequence-level encoder, so it would likely not perform as well on ASR based on a connectionist temporal classification (CTC) [18] loss, which is a token-level classification task. On top of that, if the aforementioned hypothesis regarding the relatively poor FSC performance is true, since ASR relies heavily on word detection capabilities it follows that SEGUE would perform worse.

6. Conclusion

We produced SEGUE, a pre-trained model for sequence-level SLU tasks. Our experiments demonstrated that it improves over the base wav2vec 2.0 model for many SLU tasks, some to a drastic degree, but performs worse on some tasks. We suggest that SEGUE's strength over wav2vec 2.0 is a deeper ability to understand speech, but it comes at a cost of some word detection capability. Hence, SEGUE may perform worse on tasks that rely less on understanding and more on word detection, especially when the encoder is frozen, though some of said capability may be recovered in fine-tuning. Potential future work may include using a more diverse pre-training dataset to achieve more comprehensive KD, and possibly using multilingual sentence embedders to train a multilingual utterance encoder.

7. Acknowledgment

This project is supported by the AcRF MoE Tier- 2 grant (Project no. T2MOE2008, and Grantor reference no. MOE-T2EP20220-0017) titled: "CSK-NLP: Leveraging Commonsense Knowledge for NLP", and the SRG grant id: T1SRIS19149 titled "An Affective Multimodal Dialogue System".

8. References

[1] A. Mehrish, N. Majumder, R. Bhardwaj, R. Mihalcea, and S. Poria, "A review of deep learning techniques for speech processing," 2023.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf

[3] L. Borgholt, J. D. Havtorn, M. Abdou, J. Edin, L. Maaløe, A. Søgaard, and C. Igel, "Do we still need automatic speech recognition for spoken language understanding?" 2021.

[4] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[5] W. I. Cho, D. Kwak, J. Yoon, and N. S. Kim, "Speech to text adaptation: Towards an efficient cross-modal distillation," in *Interspeech*, 2020.

[6] P. Denisov and N. T. Vu, "Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning," in *Interspeech*, 2020.

[7] S. Kim, G. Kim, S. Shin, and S. Lee, "Two-stage textual knowledge distillation for end-to-end spoken language understanding," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7463–7467.

[8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[9] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[10] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021.

[12] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: http://arxiv.org/abs/1807.03748

[13] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2236–2246. [Online]. Available: https://aclanthology.org/P18-1208

[14] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 527–536. [Online]. Available: https://aclanthology.org/P19-1050

[15] D. Gerz, P. Su, R. Kusztos, A. Mondal, M. Lis, E. Singhal, N. Mrksic, T. Wen, and I. Vulic, "Multilingual and cross-lingual intent detection from spoken data," *CoRR*, vol. abs/2104.08524, 2021. [Online]. Available: https://arxiv.org/abs/2104.08524

[16] L. Lugosch, M. Ravanelli, P. Ignoto, V. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Interspeech*, 09 2019, pp. 814–818.

[17] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," *arXiv preprint arXiv:2205.12446*, 2022. [Online]. Available: https://arxiv.org/abs/2205.12446

[18] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891