Hitachi at SemEval-2020 Task 10: Emphasis Distribution Fusion on Fine-Tuned Language Models

Gaku Morio[*], **Terufumi Morishita**[*], **Hiroaki Ozaki** and **Toshinori Miyoshi**
Hitachi, Ltd.
Resarch and Development Group
Kokubunji, Tokyo, Japan
{gaku.morio.vn, terufumi.morishita.wp,
hiroaki.ozaki.yu, toshinori.miyoshi.pd}@hitachi.com

Abstract

This paper shows our system for SemEval-2020 task 10, *Emphasis Selection for Written Text in Visual Media*. Our strategy is two-fold. First, we propose fine-tuning many pre-trained language models, predicting an emphasis probability distribution over tokens. Then, we propose stacking a trainable distribution fusion (**DISTFUSE**) system to fuse the predictions of the fine-tuned models. Experimental results show that DISTFUSE is comparable or better when compared with a naive average ensemble. As a result, we were ranked 2nd amongst 31 teams.

1 Introduction

This paper presents our strategy for SemEval-2020 task 10, Emphasis Selection for Written Text in Visual Media (Shirani et al., 2020). The task is aimed at emphasis selection, choosing candidates for emphasis in short written text in visual media (Shirani et al., 2019). Rather than predicting emphasis spans or using images, the task involves the prediction of an emphasis distribution over a short text without any image inputs.

We tackle the task by combining rich contextualized embeddings of many fine-tuned pre-trained language models (PLMs). Our strategy, shown in Figure 1, is a simple but effective meta ensemble method. First, we fine-tune heterogeneous PLMs such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), predicting an emphasis distribution over tokens. The models are trained with the KL-divergence of gold-emphasis distributions over tokens in a text fragment to handle the ambiguity within annotations (Shirani et al., 2019). Second, we propose distribution fusion (DISTFUSE) for ensembles. Different from an averaging ensemble, DISTFUSE assigns a kind of *reliability weight* to each distribution. Hence, accuracy can be improved with the method.

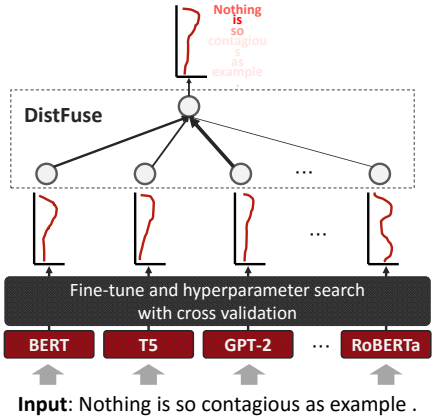


Figure 1: Overview of our strategy

We evaluate the proposed system in large-scale experiments that suggest that DISTFUSE is comparable or better when compared with the average ensemble. As a result, our system ranked 2nd amongst 31 teams. We also provide interesting insights such as on the distinct performance of PLMs, training techniques, and hyperparameters in the results section.

2 Background

The modeling of word emphasis has been widely tackled in some contexts. Zhang et al. (2016) proposed a model for extracting key phrases from Twitter text. In the context of accent, Nakajima et al. (2014)

[*]Contributed equally.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1658–1664
Barcelona, Spain (Online), December 12, 2020.

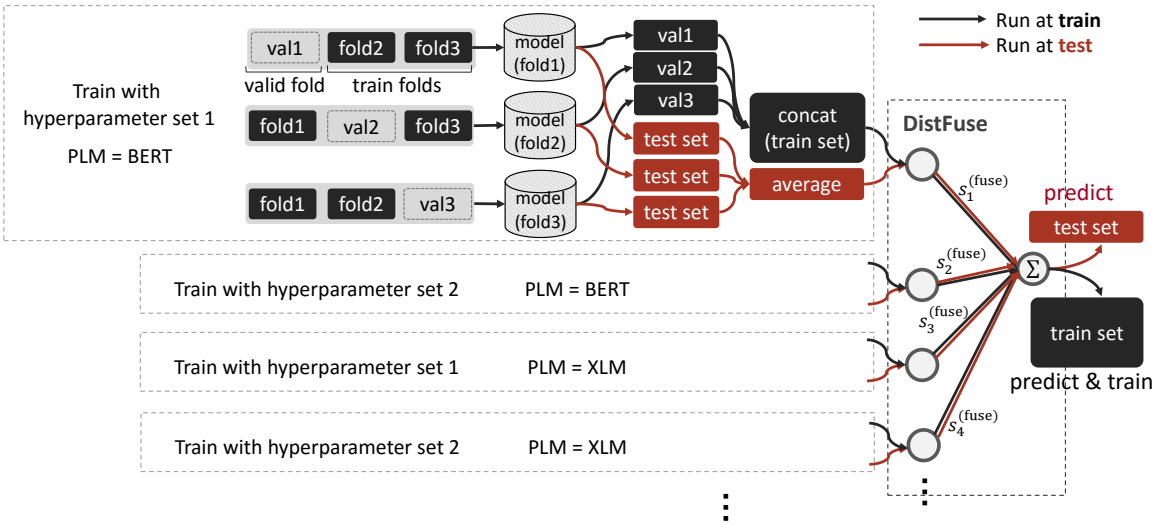


Figure 2: Overview of training and test procedures for DISTFUSE. First, we fine-tune PLMs with cross-validation and hyperparameter search. Then, we train DISTFUSE, which fuses output distributions of fine-tuned models. Black and red connections show training and test runs, respectively.

predicted emphasized phrases from Japanese advertisement text for text-to-speech synthesis. In this shared task, however, the form of emphasis is different. Shirani et al. (2019) provided a corpus with emphasized tokens based on BIO labels. The authors kept inter-subjectivity in the annotator as well as the ambiguity of the input rather than deciding gold spans. To this end, how many annotators marked a token emphasized was recorded, producing an *emphasized probability* over tokens.

3 Model

3.1 Fine-Tuning PLM for Emphasis Selection

Given a tokenized text τ, we fine-tune a PLM to predict emphasis distributions over tokens. In this study, we employ seven state-of-the-art PLMs, BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), RoBERTa (Liu et al., 2019), XLM-RoBERTa (Conneau et al., 2019), XLNet (Yang et al., 2019), XLM (Lample and Conneau, 2019), and T5 (Raffel et al., 2019). Therefore, $\texttt{PLM} \in \{\texttt{BERT}, \texttt{GPT-2}, \texttt{RoBERTa}, \texttt{XLM-RoBERTa}, \texttt{XLNet}, \texttt{XLM}, \texttt{T5}\}$. We obtain the PLM embedding of the i-th word token with a layer-wise attention (Kondratyuk and Straka, 2019):

$$\mathbf{e}^{\texttt{PLM}}_{\tau,i} = c \sum_j \texttt{PLM}_{\tau,ij} \cdot \text{softmax}\left(\mathbf{s}\right)_j,$$

where c and $\mathbf{s}$ are learnable parameters, and $\texttt{PLM}_{\tau,ij}$ is an embedding of the i-th word token[1] in the j-th layer of the PLM in the text τ. To further obtain rich features, we add part-of-speech embeddings ($\mathbf{e}^{\texttt{POS}}_{\tau,i}$) and token embeddings from a character-level LSTM ($\mathbf{e}^{\texttt{char}}_{\tau,i}$). Hence, the input representation of the i-th word token is represented as $\mathbf{e}_{\tau,i} = \mathbf{e}^{\texttt{PLM}}_{\tau,i} \oplus \mathbf{e}^{\texttt{POS}}_{\tau,i} \oplus \mathbf{e}^{\texttt{char}}_{\tau,i}$, where $\oplus$ is a concatenate operation. Then, we obtain the emphasis distribution associated with token i by simply taking feed forward networks (FFNs):

$$s_{\tau,i} = \mathbf{w}^\top \text{FFN}\left(\mathbf{e}_{\tau,i}\right) + b,$$
$$\hat{y}_{\tau,i} = \text{softmax}\left(s_{\tau,1} \ldots s_{\tau,N_\tau}\right)_i,$$

where $\mathbf{w}$ and b are learnable parameters, and N_τ is the number of tokens in the text.

Following Shirani et al. (2019), we compute the Kullback-Leibler divergence [KL-Div; (Kullback and Leibler, 1951)] for the predicted $\hat{\mathbf{y}}_\tau$ and true $\mathbf{t}_\tau$ distribution to compute the objective:

$$\mathcal{L}_\tau = D_{\text{KL}}(\mathbf{t}_\tau \parallel \hat{\mathbf{y}}_\tau) = \sum_i t_{\tau,i} \log \frac{t_{\tau,i}}{\hat{y}_{\tau,i}}.$$

[1] Sub-word tokens are averaged, and the averaged output per word token is used as $\texttt{PLM}_{ij}$.

PLM	model name
BERT (Devlin et al., 2019)	bert-large-cased-whole-word-masking
GPT-2 (Radford et al., 2019)	gpt2-medium
RoBERTa (Liu et al., 2019)	roberta-large
XLM-RoBERTa (Conneau et al., 2019)	xlm-roberta-large
XLNet (Yang et al., 2019)	xlnet-large-cased
XLM (Lample and Conneau, 2019)	xlm-mlm-en-2048
T5 (Raffel et al., 2019)	t5-large

Table 1: Provided PLMs and their type

base	value	DISTFUSE	value
folds (k)	5	optimizer	SGD
POS dim	50	initial learning rate	10.0
Char dim	50	momentum	0.9
PLM layer dropout	0.1	batch size	2048
FFN dim	200	epochs	10
FFN layer	1	fusion dropout	0.1
FFN activation	ReLU		
optimizer	Adam		
β_1, β_2	0.9, 0.999		
gradient clipping	5.0		
batch size	6		
epochs	30		

Table 2: Hyperparameter values

3.2 Cross-Validation and Model Selection

To generate better models, we fine-tune PLMs with different hyperparameter sets (e.g., learning rates and dropout ratios) as shown in Figure 2. In the training, there are four steps:

1. Generate different hyperparameter sets. For each set, provide k-fold cross validation.[2]
2. Train the model with training folds excluding the validation fold.
3. Predict the validation fold using the trained model, obtaining emphasis distributions of all training samples by concatenating the predicted validation folds. Hence, we can calculate the performance for each hyperparameter set. We select the top hyperparameter sets on the basis of the validation score.
4. Train DISTFUSE (described later) to assign a *reliability weight* to each of the top hyperparameter sets. The input for DISTFUSE is the concatenated validation folds, and DISTFUSE is trained with gold distributions.

For the test prediction, we have three steps:

1. Predict emphasis distributions of test data with the trained models for each top hyperparameter set.
2. Take an average of the output emphasis distributions for each hyperparameter set.
3. Input the averaged distributions into DISTFUSE, obtaining the final outputs.

3.3 Distribution Fusion (DISTFUSE)

To fuse the fine-tuned PLMs, we present DISTFUSE, which utilizes the meta-information of output emphasis distributions. Let $\{h_1, h_2, \dots\}$ be a set of the combinations of the top-performing hyperparameter set and PLM (e.g., $h_1 =$ (hyperparameter set 1, BERT)) and $\hat{\mathbf{d}}_{\tau,h_i} \in \mathbb{R}^{N_\tau}$ be an output emphasis distribution for h_i. DISTFUSE assigns to each distribution a kind of *reliability weight*, fusing them all:

$$\hat{\mathbf{d}}_\tau = \sum_i \hat{\mathbf{d}}_{\tau,h_i} \cdot \mathrm{softmax}\left(\mathbf{s}^{(\text{fuse})}\right)_i,$$

where $\mathbf{s}^{(\text{fuse})}$ is a tunable parameter. If $\mathrm{softmax}\left(\mathbf{s}^{(\text{fuse})}\right)_i$ is larger, the network considers the distribution $\hat{\mathbf{d}}_{\tau,h_i}$ to be more reliable, and vice versa. We also incorporate mean pooled $\hat{\mathbf{d}}_{\tau,\text{mean}}$, max pooled $\hat{\mathbf{d}}_{\tau,\text{max}}$, and min pooled $\hat{\mathbf{d}}_{\tau,\text{min}}$ distributions to the input for stable training.

Finally, KL-Div loss is employed to train DISTFUSE:

$$\mathcal{L}_\tau^{(\text{DISTFUSE})} = D_{\text{KL}}(\mathbf{t}_\tau \parallel \hat{\mathbf{d}}_\tau).$$

4 Experiments

Implementation: Seven PLMs, shown in Table 1, were provided. We implemented models with PyTorch (Paszke et al., 2019) and Hugging Face's transformer library (Wolf et al., 2019).

The learnable parameters in the models were split into two groups (Kondratyuk and Straka, 2019), one for the PLM parameters and one for all other *non* PLM parameters, assigning a different optimizer for

[2]To prevent label leakage in the fusion process, the folds are always fixed.

team	total (rank)	m=1	m=2	m=3	m=4
ERNIE	**82.3 (1)**	**72.4**	**81.9**	**86.2**	**88.7**
Hitachi (ours)	81.4 (2)	71.5	81.1	85.1	88.0
IITK	81.0 (3)	69.4	81.2	85.4	87.9
Sherry	80.5 (4)	67.7	80.3	85.8	88.1
Sattiy	79.9 (5)	67.7	79.9	85.0	87.0
baseline (BiLSTM-ELMo)	75.0 (19)	60.8	73.7	80.7	84.9

Table 3: Official scores on test set of top 5 teams and baseline. **Bold** and underline show first and second results, respectively.

	total	m=1	m=2	m=3	m=4
BERT	80.6	68.9	**81.2**	**85.3**	**87.3**
GPT-2	76.7	63.8	76.9	81.8	84.1
RoBERTa	80.2	69.9	80.4	84.4	86.3
XLM-RoBERTa	80.2	68.6	80.5	84.8	87.0
XLNet	**81.1**	**71.9**	80.8	84.8	87.0
XLM	79.6	68.9	79.4	83.8	86.1
T5	79.0	67.3	79.6	83.6	85.5

Table 4: Comparison of performance of each PLM on development set

	total	m=1	m=2	m=3	m=4
w/ DISTFUSE (ours)	**81.4**	**71.5**	**81.1**	**85.1**	**88.0**
average ensemble	81.0	70.8	80.5	85.0	87.8

(a) On test set

	total	m=1	m=2	m=3	m=4
w/ DISTFUSE (ours)	**81.8**	**71.7**	**82.0**	85.4	**88.0**
average ensemble	81.7	**71.7**	81.8	**85.6**	87.8

(b) On development set

Table 5: DISTFUSE and average ensemble comparison

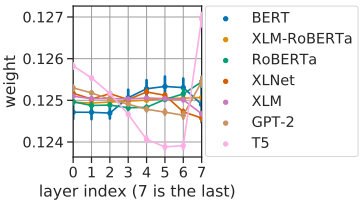


Figure 3: Weight in last eight layers for each PLM

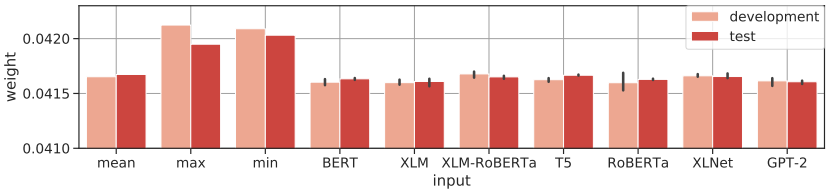


Figure 4: DISTFUSE weight (i.e., $\mathrm{softmax}\left(\mathbf{s}^{(\mathrm{fuse})}\right)$) analysis. Since we selected top 3 hyperparameter sets for each PLM, error bars of top 3 sets are presented.

each group. We froze PLM parameters for the first epoch to improve the training stability (Kondratyuk and Straka, 2019). Layer attentions were applied for the last eight layers for all PLMs, employing dropout. We applied linear warmup for learning rate scheduling with Adam (Kingma and Ba, 2015). For DISTFUSE, we employed SGD, decaying the learning rate every step.

Hyperparameter sets including learning rates and dropout ratios were generated by the Optuna framework (Akiba et al., 2019). The optimal learning rates are described in the results section. The rest of the fixed hyperparameters can be found in Table 2. We generated 40 hyperparameter sets for each PLM, and the top 3 sets for each PLM were selected for DISTFUSE.

We report the results for both the development and test sets. In the training to predict the test set, we incorporated the development set into the training set.

Metric: Systems were evaluated with Match_m (Shirani et al., 2019) defined as:

$$\mathrm{Match}_m = \frac{\sum_{x \in D_{\mathrm{test}}} |S_m^{(x)} \cap \hat{S}_m^{(x)}| / \min(m, |x|)}{|D_{\mathrm{test}}|},$$

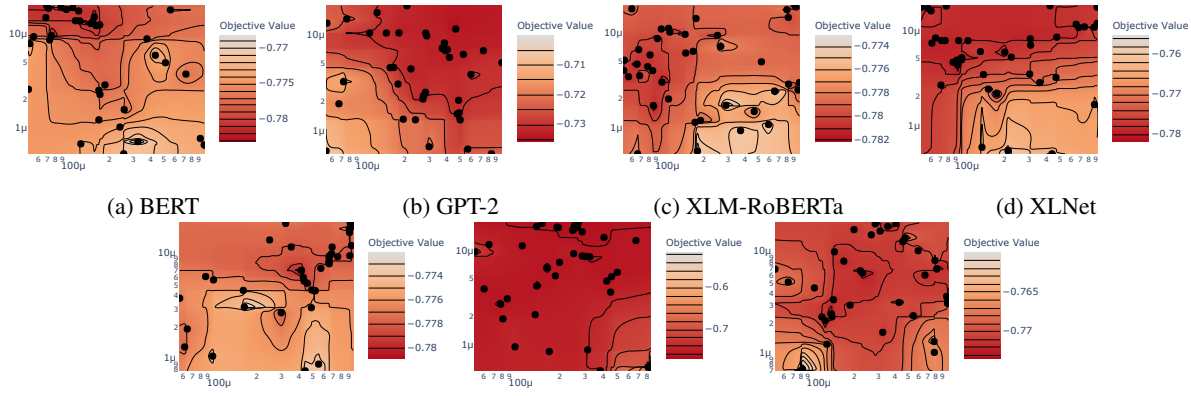
where D_{test} is the test set, $S_m^{(x)}$ is a set of $m \in \{1 \ldots 4\}$ words with top m true probabilities, and $\hat{S}_m^{(x)}$ is based on the system's top m probabilities.

4.1 Results

Table 3 presents the official test results, showing that our system is ranked 2nd. The table also shows that our system performed well when $m = 1$, implying effectiveness in detecting the most emphasized word.

When trained only on the training set without the development set, we obtained a total score of 81.2 (i.e., for each m, 71.1, 80.5, 85.3, and 88.0), showing that our model was effective even when the amount of training data was smaller.

Analyses of PLMs: To show how each PLM worked, Table 4 shows the independent performance of each PLM with the top 3 hyperparameter sets. As can be seen, the BERT and XLNet models generally performed well. Interestingly, the table also shows that the PLMs themselves were not as strong as our final model (i.e., fusing all PLM types) in most cases. This suggests that using heterogeneous PLMs can



(a) BERT (b) GPT-2 (c) XLM-RoBERTa (d) XLNet

(e) RoBERTa (f) T5 (g) XLM

Figure 5: Heatmap of negative validation scores in learning rate space, where X-axis shows learning rate for non-PLM parameters, and Y-axis shows learning rate for PLM parameters. Each point indicates searched hyperparameter set. Note that *darker colors indicate better performance*, and note that scale used for these graphs differs.

	PLM parameter	non PLM parameter
BERT	1.28×10^{-5}	1.46×10^{-4}
GPT-2	6.78×10^{-6}	3.91×10^{-4}
RoBERTa	6.95×10^{-6}	4.12×10^{-4}
XLM-RoBERTa	3.00×10^{-6}	9.92×10^{-5}
XLNet	4.29×10^{-6}	9.29×10^{-5}
XLM	6.29×10^{-6}	2.29×10^{-4}
T5	1.99×10^{-5}	2.68×10^{-4}

Table 6: Optimized learning rates

	Life	*is*	*a*	*succession*	*of*	*lessons*	..	*lived*	..	*understood*
gold	**1**	2		**1**		2				3
BERT	**1**					2				3
GPT-2	2	**1**								3
RoBERTa	2			**1**		3				
XLNet	2	3		**1**						
XLM	2	3		**1**						
XLM-RoBERTa	**1**	3		2						
T5	**1**	3								2

Table 7: Sample output of top three emphasis ranking

boost performance.

We visualized the layer-wise weight of the fine-tuned PLMs in Figure 3, showing that most weighted layers were generally found in the last several layers. However, there was a high variance, e.g., XLM and XLNet were less weighted in the last layers, and T5 had a higher up-down property.

Analyses of DISTFUSE: Table 5 compares the performance between DISTFUSE and an average ensemble. As can be seen, the proposed DISTFUSE consistently showed comparative or better performance. The result suggests that DISTFUSE is promising in terms of boosting performance.

Figure 4 illustrates the weight parameter $\mathbf{s}^{(\text{fuse})}$ of DISTFUSE, interestingly showing that the min and max pooled inputs were the most important elements. We estimate that this is because max and min pooled elements incorporate the features of the most or least emphasized information. Also, we can see that strong PLMs such as XLM-RoBERTa and XLNet were more weighted than XLM and GPT-2. We estimate that the weight assignment ability of DISTFUSE made more robust predictions.

Meta-Insights: Our in-depth analyses showed that tuning the learning rates of each PLM is important. Figure 5 visualizes the learning rate space for the two parameter groups. The figure shows that there are definitely optimal points in the learning rate for both the PLM and non-PLM groups. For example, the optimal rates of BERT were mostly found in the upper left.

We show the optimal learning rates in Table 6. XLM-RoBERTa and XLNet had relatively smaller learning rates, while BERT and T5 had larger rates. The table also shows that the learning rates of the non-PLM parameters were larger than the PLM parameters. This insight suggests that tuning the two groups independently could be effective.

Case Study: Table 7 shows an example of the output emphasis rankings for *Life is a succession of lessons which must be lived to be understood* in a validation fold. Most of the PLMs could predict that the most emphasized words would be *Life* and *lessons*, showing the promising capability of PLMs. The table also shows that each PLM had different outputs. For example, while *succession* was strongly emphasized by GPT-2, the other PLMs did not emphasize it. We can also see that some of the models

captured less emphasized tokens such as *lived* and *understand*.

5 Conclusion

In this paper, we proposed a model for the task of emphasis selection. We employed seven pre-trained language models and fused them with the distribution fusion (DISTFUSE) system. Experimental results suggested that DISTFUSE is promising in terms of boosting performance. We estimate that the effectiveness of DISTFUSE would be further validated by additional analyses (Dodge et al., 2019), which is future work. As additional future work, we will examine more effective ways of computing distributions.

Acknowledgments

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used. We thank the anonymous reviewers who gave us insightful comments. We also thank Dr. Masaaki Shimizu for the convenience afforded by these computational resources.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA. ACM.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China, November. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, November. Association for Computational Linguistics.

Solomon. Kullback and Richard A. Leibler. 1951. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hideharu Nakajima, Hideyuki Mizuno, and Sumitaka Sakauchi. 2014. Emphasized accent phrase prediction from text for advertisement text-to-speech synthesis. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 170–177, Phuket,Thailand, December. Department of Linguistics, Chulalongkorn University.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Amirreza Shirani, Franck Dernoncourt, Paul Asente, Nedim Lipka, Seokhwan Kim, Jose Echevarria, and Thamar Solorio. 2019. Learning emphasis selection for written text in visual media from crowd-sourced label distributions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1167–1172, Florence, Italy, July. Association for Computational Linguistics.

Amirreza Shirani, Franck Dernoncourt, Nedim Lipka, Paul Asente, Jose Echevarria, and Thamar Solorio. 2020. Semeval-2020 task 10: Emphasis selection for written text in visual media. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, Texas, November. Association for Computational Linguistics.