Hitachi at SemEval-2020 Task 8: Simple but Effective Modality Ensemble for Meme Emotion Recognition

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

Abstract

Users of social networking services often share their emotions via multi-modal content, usually images paired with text embedded in them. SemEval-2020 task 8, *Memotion Analysis*, aims at automatically recognizing these emotions of so-called *internet memes*. In this paper, we propose a simple but effective MODALITY ENSEMBLE that incorporates visual and textual deep-learning models, which are independently trained, rather than providing a single multi-modal joint network. To this end, we first fine-tune four pre-trained visual models (i.e., Inception-ResNet, PolyNet, SENet, and PNASNet) and four textual models (i.e., BERT, GPT-2, Transformer-XL, and XLNet). Then, we fuse their predictions with ensemble methods to effectively capture cross-modal correlations. The experiments performed on dev-set show that both visual and textual features aided each other, especially in subtask-C, and consequently, our system ranked **2nd** on subtask-C.

1 Introduction

Recently, *internet memes* — visual plus textual content on the internet — have been widely spreading due to the rapid growth of social networks, and thus, recognizing the emotions of memes is required to analyze social interactions. In SemEval-2020 task 8: *Memotion Analysis* (Sharma et al., 2020), we aim at automatically recognizing the various emotions of memes. The task contains three subtasks: **subtask-A**, where participants are required to predict the sentiment of a given meme, **subtask-B**, to predict whether a given meme represents emotions expressing certain aspects (i.e., humorous, sarcastic, offensive, and motivational) or not, and **subtask-C**, to predict a four graded degree (i.e., 0, 1, 2, or 3) to which a meme represents the emotions of the above aspects.

The challenge to deal with in the above tasks is how we can incorporate both visual and textual impressions. To this end, we propose a simple ensemble of strong pre-trained models of single modality to capture cross-modal correlations, as shown in Figure 1. To the best of our knowledge, ensembles with strong pre-trained models from different modalities have hardly been explored because multi-modal systems such as visual questions and answers (Agrawal et al., 2017) focus mostly on multimodal unified models. From this perspective, our method would provide a simple but effective approach to dealing with both visual and textual features at once.

Experimental results show that MODALITY ENSEMBLE works well for subtask-B and subtask-C, showing the effectiveness of our proposed system. The experiments performed on dev-set also show that both visual and textual features aid each other, especially in subtask-C, and consequently our system ranked **2nd** on subtask-C.

2 Background

Recent years have seen advances in the automatic recognition of visual plus textual content (Agrawal et al., 2017; Hudson and Manning, 2019). Agrawal et al. (2017) defined a multi-modal task called Vi-

[*]Contributed equally.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1126–1134
Barcelona, Spain (Online), December 12, 2020.

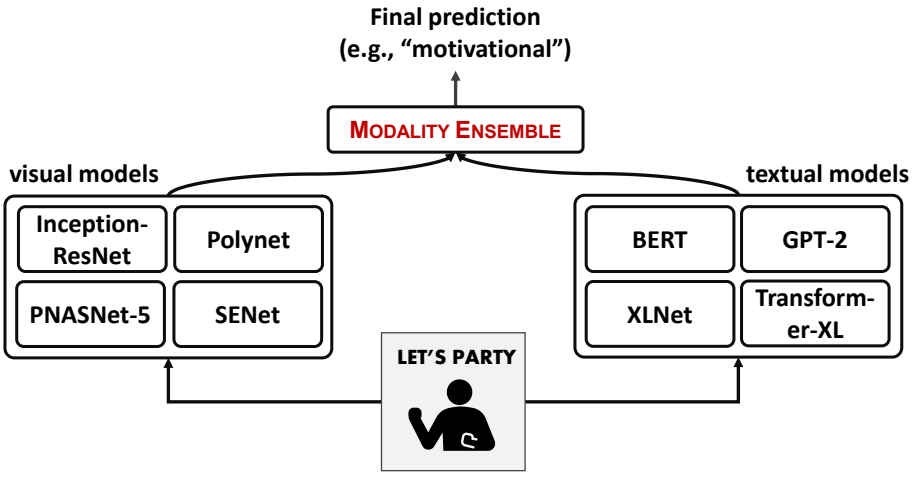

Figure 1: Schematic sketch of our method

sual Question Answering (VQA), where paired images and questions (in natural language) are supplied. Answering the questions requires an understanding of vision and language and a bit of commonsense knowledge. Given that the questions of the VQA dataset have some bias, Hudson and Manning (2019) released a new dataset called GQA, which avoids bias by automatically generating a variety of questions from scene graphs. While these tasks aim rather at understanding the *contents* of images (ex., the objects, their colors, their spatial relations, or some implications they have), Sharma et al. (2020) defined a new task, *Memotion Analysis*, aiming at automatically recognizing the *emotions* attached to the contents by the creator. We tackle this task by leveraging strong single-modal pre-trained models and fusing them to capture cross-modal correlations.

3 Task Setup

For all the subtasks, the inputs are the same, i.e., pairs of an image and a piece of text ("memes" in short).

The details of each subtask are as follows. Note that all the subtasks are classification problems on for given memes.

subtask-A is a three-class classification problem where we classify the overall sentiment into three classes, namely *negative*, *neutral* or *positive*.

subtask-B is a bundle of four binary classification problems. For a given emotion type, namely *humor*, *offensive*, *sarcasm* or *motivational*, we predict whether or not a given meme expresses the given emotion type. Note that a meme can belong to more than two emotions, so this is a multi-label classification problem.

subtask-C is a bundle of four-class classification problems for each emotion type given in the subtask-B. We classify the emotion intensity of the meme into four degrees, namely $0 =$ not, $1 =$ slight, $2 =$ normal, and $3 =$ very.

In subtask-B and subtask-C, we solved single emotion-type classification problems separately, rather than building unified models for all the emotion types.

4 Model

4.1 Overview

Figure 2 illustrates our proposed MODALITY ENSEMBLE. Given meme images, we train single-modal models (i.e., either textual or visual) for each single-emotion classification problem. Then, we just aggregate all scores of the single-modal models as the input of the ensemble models and achieve the final outputs from the models.

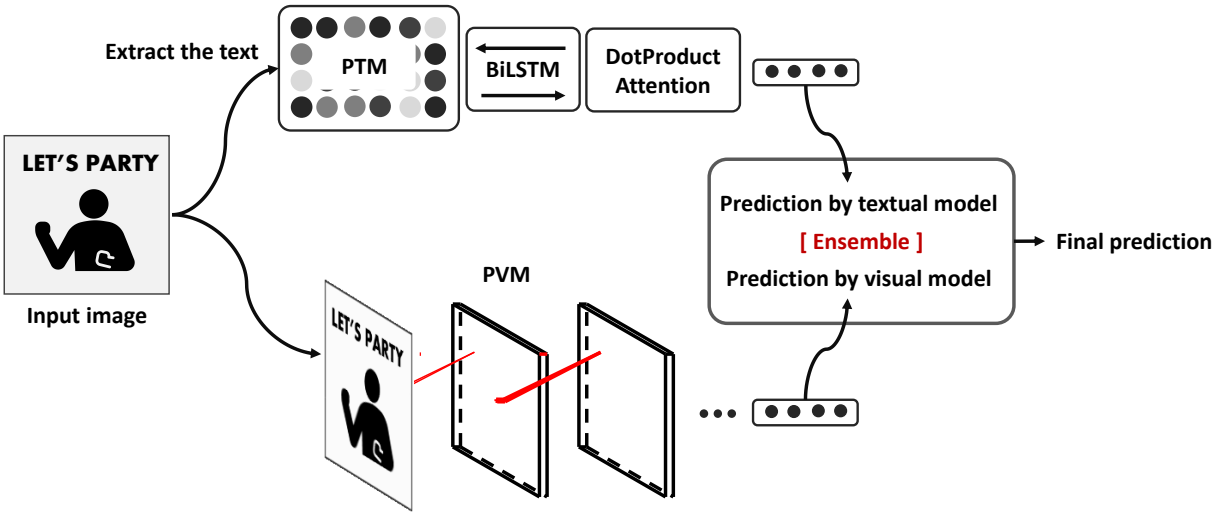

Figure 2: Overview of MODALITY ENSEMBLE. Given an input image, textual and visual models independently predict meme emotions, and the ensemble system fuses the modalities.

PVM	type	key technique
Inception-ResNet (Szegedy et al., 2016)	v2	Inception module
Polynet (Zhang et al., 2016)	-	Polynet module
SENet (Hu et al., 2018)	154 layer	Squeeze-and-Excitation
PNASNet-5 (Liu et al., 2018)	large	Sequential model-based optimization

Table 1: Four provided pre-trained visual models (PVMs)

4.2 Visual Models

We employ four types of well-known pre-trained visual models (PVMs) and fine-tune them on a given dataset. A briefly summarized list of PVMs can be found in Table 1. All these models are trained on the ImageNet dataset (Deng et al., 2009) and categorized as variations of a convolutional neural network (CNN) (Krizhevsky et al., 2012) with a residual unit (He et al., 2016) that provides shortcut connections to avoid vanishing gradients, like a recurrent neural network does. Here, we briefly summarize the four PVMs. **Inception-ResNet** (Szegedy et al., 2016) is the fusion of an Inception architecture (Szegedy et al., 2015) that incorporates convolution kernels of multiple sizes to handle the variations in the size of salient parts of images and the residual architecture. In turn, **PolyNet** (Zhang et al., 2016) provides a Polyinception module that is a polynomial combination of Inception architectures. While a residual unit in ResNet transforms the input representation $\mathbf{x}$ into $H(\mathbf{x}) = \mathbf{x} + F(\mathbf{x})$, where F is a nonlinear transformation, PolyNet pursues structural diversity for the residual unit with polynomial compositions, i.e., $H(\mathbf{x}) = \mathbf{x}+F(\mathbf{x})+F(F(\mathbf{x}))$. **SENet** (Hu et al., 2018) includes squeeze-and-excitation modules that calibrate channel-wise feature strengths by modelling correlations between channels. **PNASNet-5** (Liu et al., 2018) employs an architecture optimized by reinforcement learning and evolutionary algorithms. The core strategy is to employ sequential model-based optimization, where the authors proposed searching CNN structures in order of increasing complexity, jointly learning a surrogate model (Liu et al., 2018).

Augmentation

In the computer vision field, due to the extremely high dimensional nature of image data, augmenting training data is highly required and commonly done. We employ the following procedures for the augmentation. In the training phase, we use (i) random resizing and cropping, (ii) random horizontal flipping, and (iii) random rotation. Details on the procedure are given in Section 5.1. In the inference phase, we use "ten-crop inference" for robust prediction. This is essentially an average ensemble of the predictions on augmented images; concretely, (i) we take ten variants of images from the original image, and (ii) we calculate the log-probabilities of the classes by applying the model to all ten images. Hence,

PTM	type	key technique
BERT (Devlin et al., 2019)	large-uncased	Transformer
GPT-2 (Radford et al., 2019)	medium	Transformer encoder and decoder
Transformer-XL (Dai et al., 2019)	wt103	Inter-segment connections
XLNet (Yang et al., 2019)	large-cased	Permutation architecture

Table 2: Four provided pre-trained textual models (PTMs)

subtask-A		subtask-B					subtask-C				
label	sentiment	label	hum.	off.	sarc.	mot.	label	hum.	off.	sarc.	mot.
-1	59.6	0	76.3	60.9	77.4	35.2	0	9.2	3.1	5.4	35.2
0	31.7	1	23.7	39.1	22.6	64.8	1	32.0	21.0	22.2	64.8
1	8.7						2	35.1	36.7	49.8	-
							3	23.7	39.1	22.6	-

Table 3: Label distributions (%) in dataset. Note that hum. = *humor*, off. = *offensive*, sarc. = *sarcasm*, and mot. = *motivational*.

we get ten log-probability distributions. (iii) We average the ten log-probabilities and make predictions using the averaged log-probabilities. The ten images are made by (i) cropping four smaller images at their four corners (i.e., top-left/top-bottom/right-bottom/right-top) plus one image at its center and (ii) also getting the horizontally flipped images of the five cropped images, getting ten images in total.

Fine-Tuning

We fine-tune a PVM by replacing the top fully-connected layer of the PVM, which is used to classify original ImageNet classes, with a new one to classify the target labels of our task. During the fine-tuning, we use a single learning rate for all the layers of the model, which is common in the training of image models.

Loss Functions

We also considered the label imbalance problem. To show the importance of the problem, we show Table 3 with the number of samples for each class. For example, label "1" in subtask-A is 8.7%, and "0-off." in subtask-C is only 3.1%, showing that the numbers of samples belonging to the classes are highly imbalanced. Therefore, we employ class-wise weighted loss where the weight for each class is proportional to the inverse of the number of samples belonging to that class.

4.3 Textual Models

We employ four types of pre-trained textual models (PTMs). Brief summarized explanations of each PTM can be found in Table 2. All these models are based on a Transformer (Vaswani et al., 2017) language model, which stacks layers of multi-head self-attentions. The differences between PTMs are as follows. **BERT** (Devlin et al., 2019) is a bidirectional Transformer trained by masked language modeling and sentence prediction. Although there are some variant pre-trained models of BERT, we selected a large model for higher performance. **GPT-2** (Radford et al., 2019) employs a Transformer encoder and decoder trained by left-to-right language modeling. **Transformer-XL** (Dai et al., 2019) also contains a Transformer encoder and decoder trained by left-to-right language modeling with inter-segment connections to capture longer dependencies. **XLNet** (Yang et al., 2019) is a Transformer-based model but incorporates training on permutations of gold tokens to incorporate bidirectional contexts without corrupting the original tokens with mask tokens.

Preprocessing

Text in memes is often in upper-case characters. We normalize the characters by converting them to lower-case characters. After the conversion, we tokenize the text with PTM-specific tokenizers (see Section 5.1 for details).

	value
optimizer	SGD
momentum	0.95
learning rate scheduling	$\times 0.1$ when epoch reaches 40 and 60
learning rate	The best one from [1e-4, 2e-4, 5e-4, 1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2]
training epochs	100
batch size	8 to 32 depending on `PVM`
resized image size	given by each `PVMs`
FFN dim	Induced by resized image size and the pre-trained model architecture

Table 4: Hyperparameters of visual models

	value
folds (k)	5
batch size	1 to 16 depends on `PTM`
gradient clipping	5.0
early stopping patience	5
max epochs	30

(a) Fixed hyperparameters

	value range
learning rate	1e-6 to 3e-4
optimizer	Adam or Adam with warmup
dropout	0.0 to 0.3
BiLSTM layers	[1, 2]
BiLSTM dim	[256, 512, 1024]
attention dim	[256, 512, 1024]
FFN dim	[256, 512, 1024]

(b) Optimized hyperparameters

Table 5: Hyperparameters of textual models

Fine-Tuning

We fine-tune four `PTMs` as mentioned above with some additional task-specific layers for single-emotion classification tasks.

First, we feed tokenized pieces of text into `PTM` to get context-specific embeddings. We also apply a bidirectional LSTM (BiLSTM) (Graves et al., 2013) and dot-product attention to further contextualize the embeddings. To produce a sentence representation, we apply `PTM`-specific pooling, which takes the last embedding for GPT-2 and XLNet, takes the first (i.e., [CLS]) embedding for BERT, or takes a max-pooling for Transformer-XL. Finally, the embedding is fed into an FFN to predict the class label. We use the weighted cross-entropy loss, the same as the one shown in Section 4.2.

4.4 Modality Ensemble

Our MODALITY ENSEMBLE fuses outputs of fine-tuned `PVMs` and `PTMs` to capture cross-modal correlations. We employ stacked generalization (Wolpert, 1992), one of the ensemble methods, as well as naive average ensemble methods. Stacked generalization employs a meta-estimator (e.g., a simple linear model), which aggregates the predictions of base models to make more robust predictions.

Although mostly linear models are utilized, we hypothesized that *non-linearity* may be essential for capturing complicated correlations of modality predictions, so we tried several non-linear estimators (ex., decision tree and random forest) as well as linear estimators like logistic regression.

5 Experiments

5.1 Implementation

For the implementation of the visual models, we used mainly the Torchvision (`https://github.com/pytorch/vision`) and Pillow (`https://github.com/python-pillow/Pillow`) libraries for preprocessing. We used the `RandomResizedCrop()`, `RandomHorizontalFlip()`, `TenCrop()`, and `RandomRotation()` functions of the Torchvision library with their default parameters for augmenting the images. To fine-tune `PVMs`, we used the cnn_finetune (`https://github.com/creafz/pytorch-cnn-finetune`) library, which in turn utilizes pre-trained models.

subtask-A		subtask-B		subtask-C	
IITK (1)	.355	vlad eduardgzaharia UPB (1)	.518	guoym (1)	.322
guoym (2)	.352	guoym (2)	.515	**Hitachi** (2)	.319
aihaihara (3)	.350	Kraken (3)	.510	vlad eduardgzaharia UPB (3)	.317
Diptadas (4)	.349	upv (4)	.509	ripple ai (4)	.316
IrinaBejan (5)	.348	memebusters (5)	.509	IITK (5)	.315
Hitachi (15)	.341	**Hitachi** (21)	.491		

Table 6: Official results of average macro-F performance (and its rank) for top five teams.

		subtask-A	subtask-B					subtask-C				
			ave.	hum.	off.	sarc.	mot.	ave.	hum.	off.	sarc.	mot.
ensemble all (MODALITYENSEMBLE)		.371	**.540**	.556	**.548**	.532	.526	**.338**	.278	**.276**	**.268**	.526
PVM	ensemble (vision)	.357	.526	.527	.526	.526	.524	.334	**.280**	.274	.258	.524
	PNASNet-5	.331	.514	.526	.515	.507	.508	.324	.279	.252	.247	.508
	Inception-ResNet	.342	.509	.524	.509	.499	.504	.321	.275	.268	.245	.504
	SENet	.308	.489	.509	.478	.481	.488	.307	.255	.263	.233	.488
	Polynet	.329	.498	.505	.498	.488	.503	.317	.263	.247	.245	.503
PTM	ensemble (text)	**.374**	.535	.544	.528	**.536**	**.531**	.331	.270	.262	.265	**.531**
	BERT	.364	.536	**.567**	.515	.532	.530	.326	.271	.241	.260	.530
	GPT-2	.358	.529	.558	.521	.522	.512	.298	.250	.215	.224	.512
	Transformer-XL	.346	.523	.539	.521	.528	.504	.304	.260	.231	.213	.504
	XLNet	.358	.515	.522	.522	.511	.504	.305	.241	.232	.232	.504

Table 7: Modality ablation study on dev-set. Macro-F score of single emotion classification and average scores (=ave.) are shown. Note that mot. in subtask-B and subtask-C shares same scores since it is binary classification in both subtasks.

For the implementation of the textual models, we employed Jiant (Pruksachatkun et al., 2020), a transfer learning framework that incorporates Hugging Face's transformer library (Wolf et al., 2019) for `PTMs` and tokenizers.

Some of the other codes were built with PyTorch (Paszke et al., 2019) and Ignite (`https://github.com/pytorch/ignite`). For the meta-estimators, we tried classifiers like logistic regression, decision tree, and hard/soft-voting and chose the one that performed the best in our preliminary experiments. The meta-estimators were implemented with scikit-learn (Pedregosa et al., 2011).

5.2 Hyperparameters

For the visual models, we searched hyperparameter space with a relatively small number of fixed values because the training cost is much higher than that of textual models. The hyperparameter range for the visual models is shown in Table 4.

For the textual models, we optimized hyperparameters as shown in Table 5b. The hyperparameter search was conducted by using Optuna (Akiba et al., 2019), an optimization framework, in 30 steps. The fixed hyperparameter ranges for the textual models are shown in Table 5b. During the hyperparameter optimization, the performances were measured by 5-fold cross-validation.

5.3 Results

Official Ranking

First, we report the official scores and ranking in Table 6. The table shows that our system was ranked 2nd in subtask-C, showing the effectiveness of our system.

subtask-A	subtask-B				subtask-C			
	hum.	off.	sarc.	mot.	hum.	off.	sarc.	mot.
HV	DT	RF	DT	DT	HV	DT	DT	SV

Table 8: Best ensemble methods for MODALITY ENSEMBLE. HV, SV, RF, and DT denote hard vote, soft vote, random forest, and decision tree, respectively.

Analyses of Modality Ensemble

We show an ablation study on the dev-set in Table 7. In the study, we examined the performances of single PVMs and PTMs, ensemble of models from single modalities ["ensemble (vision)" and "ensemble (text)"], and ensemble of models from all modalities (MODALITY ENSEMBLE). As can be seen from the table, in most tasks, MODALITY ENSEMBLE performed better than or was at least comparable to single-modal ensemble models. These results suggest the effectiveness of MODALITY ENSEMBLE. This would be because MODALITY ENSEMBLE successfully captures the correlation of cross-modal predictions.

In subtask-A, the text-only ensemble models performed the best among all the ensemble models, outperforming MODALITY ENSEMBLE and the vision-only ensemble models. In addition to this, single textual models often performed better than single visual models. This implies the superiority of the textual model to the visual model for the sentiment classification task.

In subtask-B, MODALITY ENSEMBLE performed the best on average, outperforming the vision-only or the text-only ensemble models. For single modal models, generally, the textual models outperformed the visual models. This also implies the superiority of textual modality in binary emotion classification tasks.

In subtask-C, MODALITY ENSEMBLE performed the best on average, followed by vision-only ensemble models and textual-only ensemble models. The same tendency was seen in the comparison of single models. This tendency is in contrast to that of subtask-B, implying the superiority of visual modality in emotion grading tasks.

In terms of PVMs, PNASNet and Inception-ResNet worked well generally, although the two models are came before SENet and PNASNet. For the PTMs, BERT is likely the best model. However, we estimate that more hyperparameter optimizations could improve the weaker PVMs and PTMs.

Which Meta-Estimator Is the Best?

Table 8 shows the best meta-estimator for each emotion classification task. In most emotion classification tasks, the non-linear ensemble methods performed the best. We guess that complicated cross-modal correlations are better captured by non-linear methods.

6 Conclusion

In this paper, we presented a simple but effective modality ensemble for predicting multi-modal internet meme emotions. For both visual and textual modalities, we fine-tuned strong pre-trained models independently. In addition, we fused the predictions with an ensemble method to capture cross-modal correlations. The experiments on the dev-set show the promising results of our strategy. We will explore a more effective way of handling the multi-modality of an internet meme.

Acknowledgments

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used. We thank Dr. Masaaki Shimizu for the convenience of computational resources.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2017. VQA: Visual question answering. *International Journal of Computer Vision. J. Comput. Vision*, 123(1):4–31, May.

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.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July. Association for Computational Linguistics.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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.

Alex. Graves, Abdel rahman. Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. 2018. Progressive neural architecture search. In *The European Conference on Computer Vision (ECCV)*, September.

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.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Yada Pruksachatkun, Phil Yeres, Haokun Liu, Jason Phang, Phu Mon Htut, Alex Wang, Ian Tenney, and Samuel R. Bowman. 2020. jiant: A software toolkit for research on general-purpose text understanding models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 109–117, Online, July. Association for Computational Linguistics.

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

Chhavi Sharma, Deepesh Bhageria, William Paka, Scott, Srinivas P Y K L, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 Task 8: Memotion Analysis-The Visuo-Lingual Metaphor! In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, Sep. Association for Computational Linguistics.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex A. Alemi. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR) 2016 Workshop*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

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.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.

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.

Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. 2016. Polynet: A pursuit of structural diversity in very deep networks. *arXiv preprint arXiv:1611.05725*.