StoryReel: Generating Short Stories From Celebrity Images

Arushi Jain, Sagnik Sinha Roy
Data Science, School of Information
University of Michigan Ann Arbor
arushij@umich.edu, sagniksr@umich.edu

Abstract

We provide a model which takes an image as an input, and combines multiple neural models for face recognition, object detection and text generation to generate a relevant short story. Tasks like retrieval from a database and named entity recognition are also included in this model. The model identifies celebrities present in the image along with any objects, and returns a thematic short story based on them. The final model consists of 5 components: Face recognition, object detection, retrieval, named entity recognition and text generation. The first 2 components function independently, and the output of each subsequent component becomes the input for the next. We performed qualitative analysis on the results, and found the outputs to be quite promising.

1 Introduction

Researchers have made significant progress in advancing artificial intelligence since Turing famously asked "Can machines think?" Machine learning techniques have been consistently improving efficiency in most of the industries [4]. Many labour tasks have become automated or "smarter" from the use of machine learning algorithms.

Deep learning models have shown a huge success in a variety of fields in recent years, with most of it coming from their ability to automate the frequently tedious and difficult feature engineering phase by learning "hierarchical feature extractors" from data [5]. They have enabled remarkable progress over the last years on a variety of tasks, such as image recognition, speech recognition, and machine translation. One crucial aspect for this progress are novel neural architectures. Currently employed architectures have mostly been developed manually by human experts, which is a time-consuming and error-prone process.

Talking about Natural language processing in particular, there has been a huge amount of improvement in the last decade, with the advent of openly available datasets, and powerful machines. Tasks like named entity recognition, sentiment analysis, etc have taken over the industry by storm, and are used in a wide variety of applications.

Through this project, we aimed to discover the potential of machines in the creative writing industry. While machines are adept at performing procedural tasks, can they truly identify different parts of image and simulate the human creativity based on the elements of that image? This is the question we aimed to answer satisfactorily by developing a fictional story completion algorithm from celebrity images. An algorithm that can create fictional stories close to human writing would be revolutionary, and it would be very interesting to see the pieces of text that machines could potentially come up with based on input images.

In this project we combined various state-of-the-art neural models for multiple tasks together: face detection, face recognition, object detection, and text generation, and also use named entity recognition and retrieval in the model. While combination of multiple neural models can be tricky due to the exponential increase of errors in the different steps, we take several error minimizing

strategies to ensure that the generated stories are highly relevant to the given image. We present some interesting results of our model in the Results section, and found that the fictional stories generated by our model were in general, really accurate in context to the input image.

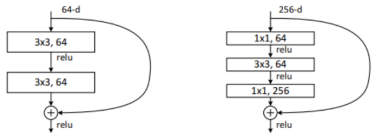
2 Related Work

2.1 Face Recognition

Face recognition has the perception of a solved problem, however when tested at the million-scale exhibits dramatic variation in accuracies across the different algorithms. VGG face is a very renowned dataset when it comes to face recognition as it is created from celebrity faces. Usually private sets, e.g.,Google, Facebook, or governmental databases can not be made public and private data can get to as many as 8M identities and 200M+ photos while the largest public dataset has 100K identities and 10M photos [6]. That is the reason people usually use celebrity images to develop face recognition algorithms. Although by training only on celebrity photographs, we risk constructing a bias to particular photograph settings. For example, it is reasonable to assume that many celebrity photographs were obtained with high quality professional cameras, or that many celebrities photographs are not of children. Algorithms learned with this bias may perform differently when tested on photographs of non-celebrities [12].

Another challenging task here is large scale labeling of such datasets. Since labeling million-scale data manually is challenging and while useful for development of algorithms, there are almost no approaches on how to do it while controlling costs. Even big companies like MobileEye, Tesla, Facebook, hire thousands of human labelers, costing millions of dollars. But it has been seen that large-scale training optimization considers large numbers of samples per class where batching and online approaches like stochastic gradient descent, are valuable. Techniques like data augmentation are really helpful such as expression altering or pose warping as then samples can be parametrized and trained more effectively. This leads to another challenging task which is how to scale and optimize training in case the number of classes (rather than samples in each class) is big.

ResNets have achieved impressive, record-breaking performance on ImageNet dataset. Therefore, in this paper [7], particularly we used ResNet50 for face recognition. ResNet revolutionized the field of image classification using CNNs by solving the "vanishing gradient" problem, and thus, introducing a way to train very deep models. The key idea behind this solution is the concept of "identity shortcut connection", where the nodes in the network sometimes skip one, or even multiple layers, as shown in the following diagram:



Another concept that was used in this model was the idea of "residual learning", where basically the model tries to approximate the activation functions between layers, which leads to a reduced complexity for the model. So, the residual block was refined, and a pre-activation variant of the residual block was used, which led to the gradients in the network flowing through these shortcut paths. These techniques can be used to effectively train even a 1000-layer neural network.

Although the authors included several variants of the model. They had the basic 18-layered and 34 layered models. To prove the effectiveness of the techniques, they even included larger models, like 50 layers, 100 layers and even 1000 layer. We used the variant ResNet-50, which is a 50 layer deep neural network. These layers include many 3x3 convolution layers, max-pooling layers, etc.

2.2 Object Detection

The modern history of object recognition goes along with the development of ConvNets, which was all started here in 2012 when AlexNet won the ILSVRC 2012 by a large margin. AlexNet bases on the decades-old LeNet, combined with data augmentation, ReLU, dropout, and GPU implementation.

It proved the effectiveness of ConvNet, kicked off its glorious comeback, and opened a new era for computer vision. Then ZFNet, VGGNet and Inception models after that. We specifically used VGG-16 for object detection in this paper. The main idea of VGGnet[19] is to replace large-kernel conv by stacking several small-kernel convs. It strictly uses 3x3 conv with stride and padding of 1, along with 2x2 max-pooling layers with stride 2. In particular, an important role in the advance of deep visual recognition architectures has been played by the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)[16], which has served as a testbed for a few generations of large-scale image classification systems, from high-dimensional shallow feature encodings [13] (the winner of ILSVRC-2011) to deep ConvNets [8](the winner of ILSVRC-2012).

2.3 Retrieval

It has been shown that semantic similarity tasks do not accurately measure the effectiveness of an embedding in the other down-stream tasks ([17], [21]). Furthermore, human annotation of similarity at sentence level without any underlying context can be subjective, resulting in lower inter-annotator agreement and hence a less reliable evaluation method. There has not been any standardized intrinsic evaluation for the quality of sentence and document level vector representations beyond textual similarity. There is therefore a crucial need for new ways of evaluating semantic representations of language which capture other linguistic phenomena.

Although, there are many methods which could have been used for retrieval of top stories a relevant story from a database, given the objects present in the image. But we specifically used sentence vectorization approach inspired by the paper by ''Story Cloze Evaluator: Vector Space Representation Evaluation by Predicting What Happens Next" [11]. They basically calculate the average vector representation for the paragraphs/sentences and match with the context based on cosine similarity.

2.4 Text Generation

Recurrent neural networks (RNNs) has been the most popular way to solve the problem of text generation, or text completion. But since the inception of Transformer based Networks, people have started experimenting it technique for text generation. Not surprisingly, it is producing far better results than LSTMs. There are many papers which have utilised RNNs to perform story completion.The Effect of Different Writing Tasks on Linguistic Style: A Case Study of the ROC Story Cloze Tasks [18] used logistic regression after generation to categorize an ending, either as right vs. wrong or as original vs. new. Writing Stories with Help from Recurrent Neural Networks [15] uses the evaluation functionality of Creative Help to compare RNNs to alternative approaches such as the described case-based reasoning method. Incorporating Structured Commonsense Knowledge in Story Completion [2] uses three types of information: narrative sequence, sentiment evolution and commonsense knowledge for story completion.

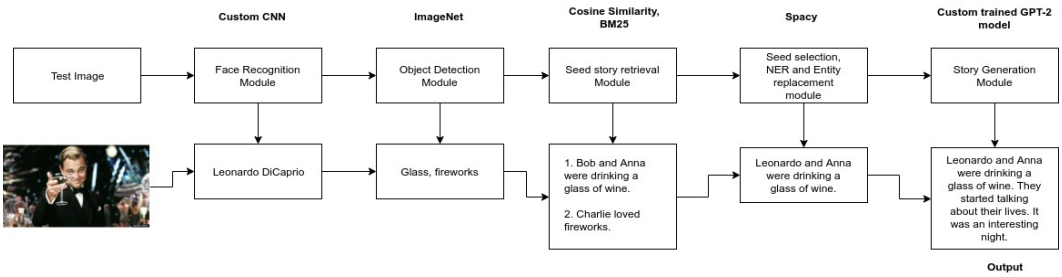
But Google launched one paper called Attention Is All You Need [22] in 2018 where they propose Transformer Networks based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Basis of BERT and GPT-2 both is actually attention-based mechanisms. Before the release of BERT there was GPT model(previous version of GPT-2) by OpenAI which trains a left-to right Transformer LM on a large text corpus. Infact, many of the design decisions in BERT were intentionally chosen to be as close to GPT as possible so that the two methods could be minimally compared. In Feb 2019, OpenAI published another paper called Language Models are Unsupervised Multitask Learners [14] introducing GPT-2 which is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting. Samples from the model reflect improvements and contain coherent paragraphs of text which was also true in our case when we tried to predict paragraphs of text using Essays and Game of Thrones corpora. On reading comprehension the performance of GPT-2 is competitive with supervised baselines in a zeroshot setting. However, on other tasks such as summarization, while it is qualitatively performing the task, its performance is still only rudimentary according to quantitative metrics.

3 Approaches

The flow of the model is visualized in the diagram below. From the input image, the face recognition model and object detection model work independently to extract the celebrities and objects present

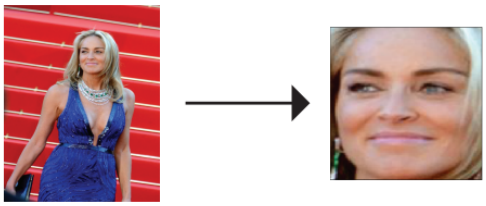
in the image respectively. We limit the results to one celebrity and three objects with the highest probabilities so as to minimize propagation of error. The output from the object detection module is used in the retrieval module to retrieve relevant stories from our database of short stories. The retrieval module just retrieves one sentence, which would be the starting sentence for the final generated story. The output from the retrieval module and the face recognition module is sent to the NER module. Here, it tries to find the named entities in the selected story, that are tagged as "PERSON". The first such occurrence is replaced with the first name of the celebrity found by the face recognition module. We do the replacement process early so as to minimize error and prevent propagation, similar to last time. Finally, the output of this model is sent to the fine-tuned GPT-2 model for generating the resulting short story. The following sections elaborate more on each individual modules.

Data Flow Diagram



3.1 Face Recognition

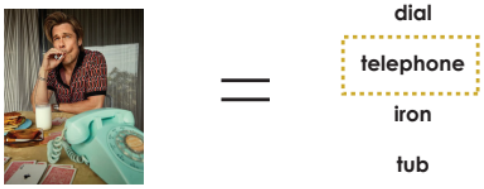
The first step in our process involves recognizing faces in the images. To build the baseline model, we implemented the “Local Binary Patterns Histograms” (LBPH) algorithm [23]. LBP is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. This has proven to be a very effective algorithm for texture classification. Making histograms out of this can represent images of in a single vector. We used OpenCV to extract the face out of an image, and then used the LBPH algorithm to train our model. For the main model, however, we used the MTCNN architecture [24] for the task of detecting faces. For face recognition, we used the ResNet-50 model [7], with pre-trained weights on the VGGFace2 dataset [1]. The implementation of VGGFace2 in Keras provides pretrained weights on 3 different models: VGG16, ResNet-50 and SeNet. We chose ResNet as we have worked with it before, in terms of implementation all 3 models have similar performance on the VGGFace2 dataset. We didn’t try to finetune the weights of this model as the VGGFace2 dataset comes with 9000+ celebrities with over 3.3 million faces, and we didn’t have the computational resources necessary to train such a model. Applying this network to a test image provides a list of identified celebrities with the prediction probabilities for each. We select the face with the highest prediction probability. This adds the limitation that our model will only generate a story about one face for a given image, but we felt that reducing errors is a bigger priority.



3.2 Object Detection

The next step in our process was identifying objects from the submitted images. We use the VGG-16 model [19] for object detection. The pre-trained weights implemented in Keras can classify images into upto 1000 different classes, which are trained on the ImageNet dataset [16]. Keras also provides the ability to choose different CNN types to perform the classification. We chose the VGG-16 model as it is a lightweight model which provides a good accuracy. We also experimented with Xception

[3], VGG-19 [19], and Inception-v3 [20], but we felt the object detection provided by VGG-16 was the best. We did not perform any fine-tuning on the weights. Applying this model on a test image again provides a list of detected objects in the image with the prediction probabilities. We select the top 3 objects for passing into the retrieval module.



3.3 Retrieval

The third step in our process was to retrieve a relevant story from a database, given the objects present in the image. We prepared a database of short stories using a portion of the ROCStories corpus [10]. This dataset contains 3-4 sentence stories. The dataset was split into sentences, and given the objects found in the image, we performed a search on the first sentence of the dataset to find words similar to the theme of the objects found in the previous step. We achieved this by using Google's Word2Vec [9] to approximate a vector out of the first sentence of each of the records. The object's word embedding is then compared to these approximated sentence vectors, and the closest one is retrieved. This methodology ensures that the stories retrieved are in a similar theme to the objects found in the image, while not being limited by a direct word search. So, even if an object found is not present in our dataset, the algorithm should be able to retrieve a closely related story. This model just returns one sentence which will be used as a seed sentence in the final step.

3.4 Named Entity Recognition

The penultimate step of the module is to replace the relevant entities in the seed sentence extracted from the last step. We have browsed over the ROCStories corpus, and most stories have named entities in the first sentence who serve as the subjects of the story. We use Spacy NER to extract the named entities in the sentence tagged as the label "PERSON" by the sentence tokenizer. We then replace this entity with the name of the person extracted in step 1. Since the first sentence is short, this ensures that the output of the face recognition model is inserted into the story right into the beginning, which reduces chances of errors. If we were to replace the entity after generation of text, the output may be more prone to errors. The modified seed is then prepared as the input for the final generation text.

3.5 Text Generation

The final step in the process is to generate a short story from the seed text. Historically, LSTMs were the default strategy for text generation. Ever since the introduction of transformer networks however, there has been a massive shift in text generation. We used the GPT-2 model [14] released by OpenAI. The model we used had 345 million parameters. We finetuned the model on the remaining portion of the ROCStories dataset [10], and trained the model for a 1000 epochs. This led to really good generation results from our model. GPT-2 has the property of adapting well to small datasets on finetuning, and we saw this behavior in this case.

4 Experiments

4.1 Data

4.1.1 Face Recognition

As mentioned in approach, for face recognition, we used the ResNet-50 model, with pre-trained weights on the VGGFace2 dataset [1]. VGGFace2 is a large-scale face recognition dataset. Images are downloaded from Google Image Search and have large variations in pose, age, illumination, ethnicity

and profession. It contains images from identities spanning a wide range of different ethnicities, accents, professions and ages. All face images are captured "in the wild", with pose and emotion variations and different lighting and occlusion conditions and Face distribution for different identities is varied, from 87 to 843, with an average of 362 images for each subject. This dataset was collected with three goals in mind: (i) to have both a large number of identities and also a large number of images for each identity; (ii) to cover a large range of pose, age and ethnicity; and (iii) to minimise the label noise. We describe how the dataset was collected, in particular the automated and manual filtering stages to ensure a high accuracy for the images of each identity [1].

4.1.2 Object Detection

As mentioned above, we used pre-trained weights of VGG-16 model for object detection that can classify images into 1000 different classes. These pre-trained models are on ImageNet dataset. It is a very well known dataset, aimed at (manually) labeling and categorizing images into almost 22,000 separate object categories for the purpose of computer vision research which later turned into a ImageNet Large Scale Visual Recognition Challenge, or ILSVRC for short [16]. The goal of this image classification challenge is to train a model that can correctly classify an input image into 1,000 separate object categories. Models are trained on 1.2 million training images with another 50,000 images for validation and 100,000 images for testing. These 1,000 image categories represent object classes that we encounter in our day-to-day lives, such as species of dogs, cats, various household objects, vehicle types, and much more. When it comes to image classification, the ImageNet challenge is the de facto benchmark for computer vision classification algorithms, that is the reason we used this for object detection purposes.

4.1.3 Retrieval

We used ROC short stories dataset [10] for retrieving the most relevant story based on the object detected in the image. This dataset was released by University of Rochester which contains approximately 90,000 five-sentences stories with a title. It was released with the challenge to predict right endings, given first four sentences of the story but concatenated the right endings and used it for retrieving the most similar sentence.

4.1.4 Text Generation

We fine-tuned GPT2 model on ROC short stories dataset to generate the stories. GPT2 model is trained on WebText [14] which was prepared by scraping web pages that have been curated/filtered by humans and contains the text subset of these 45 million links.

4.2 Evaluation and Results

Since we used pre-trained weights for face-recognition and object detection, we only evaluated the text generation with GPT-2 model. However, we report the standard evaluation results for each of the models we used. The metric used for evaluating the ResNet model on the VGGFace dataset is TAR @ FAR (True Acceptance Rate at False Acceptance Rate), which is a standard metric used for classification of images. The ResNet-50 model has a TAR @ FAR score of 0.891 at the 0.001 significance level. We analyzed the results of this model qualitatively by running the face detection tasks as well as the face recognition on random images of celebrities on the web. We saw an excellent performance of these models. The face was cropped perfectly and it was correctly classified to the correct celebrity in all of the cases we tried, with prediction probabilities of over 0.85.

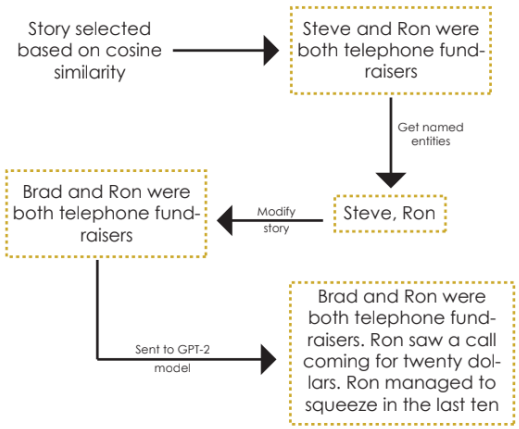
Next, for the object detection model, as mentioned before, we used the VGG-16 model which was trained on the ImageNet dataset. This model has a top-5 test error rate of just 6.8%. We again analyzed the model qualitatively. The model returns around 4-5 objects found in a given test image, along with the prediction probabilities. We observed that the first 2-3 objects (in descending order of prediction probabilities) found were very accurate with respect to the image. The last few objects were not really relevant to the picture, for eg. a telephone was identified as an iron due to similar look. However, using the first 2 results out of this model is a good idea to get relevant stories.

For retrieval, we found that the word embedding representation and cosine similarity retrieval is a surprisingly good solution to the problem which is applicable to small datasets. For example, in one of the images, the object found was gown, and in our story database, there wasn't any story with the

word “bikini” in it. However, there was one with “swimsuit”, and this was returned by the retrieval model, which led to a story relevant to the image even with the limited vocabulary of our database.

To evaluate the generated text, we implemented a GAN based discriminative classifier. For the stories in the test dataset of the ROCStories corpus that our custom GPT-2 model was trained on, we created a dataset of 50% human written stories, and 50% GPT-2 generated stories (1 human written story and 1 GPT-2 generated story for each seed node). From this test dataset, we created a TF-IDF based feature vector, and trained a Logistic Regression classifier for this model to classify the human written stories from the GPT generated ones. The goal for this classifier is to have an accuracy or F-1 close to 0.5, which would imply that the stories generated by the GPT is indistinguishable by humans for the classifier. Our classifier achieved an accuracy of 0.67 and an F-1 score of 0.63. We also found the cosine similarity scores of the average vector which represents the actual human continuation with the average vector which represents the generated text to see whether the generated text falls in the same theme as desired. The average cosine similarity for all the documents in the test corpus was 0.54.

Flow of Results from different Parts



Final Results with Different Celebrity Images

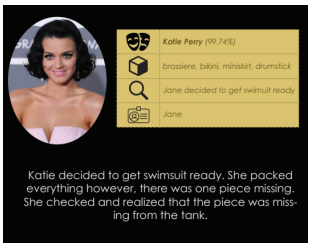


Figure 1: Katie Perry

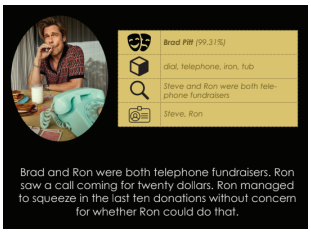


Figure 2: Brad Pitt

Figure 3: Channing Tatum

Figure 4: Sharon Stone

5 Discussion

All four figures shown above in Results section perfectly predicts the name of the celebrities but we saw some examples where model was not able to predict the correct celebrity name. In some cases, we saw that it gives higher probability to some other celebrity name but not one in the image, this could be because of the fact that some images differ in the angle and lighting which leads to predicting some other celebrities' names. This can be improved more by fine-tuning on a smaller dataset which could accurately learn facial features, light settings etc and then predict. It is also one of the limitations of our model since we used the celebrity name with highest predicted probability for next tasks in the pipeline.

Our object detection model also returns list of detected objects in the image with the prediction probabilities where we particularly saw that sometimes it detects objects which are not present in the image or which resemble to objects present in the image. For example:- it predicts miniskirt, cardigan whereas she was wearing a evening gown which was also the part of list. Here, we took top 3 objects detected for passing into the retrieval module and retrieve the relevant story.

While retrieving the relevant story from the ROC short-stories dataset, we use cosine similarity so it was seen that sometimes it does not take the context into account and retrieved story might not be the best fit. We can further improve in this by taking into account the context of the object in the image and then retrieve the relevant story.

In our NER model, where we replace the name of the celebrity with one of the names in the story using Spacy module also had some limitations. First, spacy sometimes doesn't classify the names as entities if they are not capitalised and second our model currently doesn't replace the gender in the retrieved story that is if the image we inputted is of Sharon Stone but story retrieved is on a male so current model does not replace he to she or takes into account the gender of the celebrity. Although this is an easy fix but we were not able to do it due to time constraints but there is a annotated dataset which tells the gender of celebrities which are present in VGG Face2 dataset, so we can use that in future to improve this aspect.

Lastly, we saw that generated short stories were pretty great and are the highlight of our paper. As one can see, all the different generated short-stories in the results above highlight the power of Transformer based language generation models. One fascinating observation made by us was that the length and format of the text varied greatly based on the fine tuning. For the short stories dataset, length of generated sentences were about four to five words, and the story was switched in three or four sentences, which is exactly like the corpus. For short stories, it was hard to identify that they were generated by machines on a quick look. We can say that for the generated short stories, it is able to logically continue the input seed and satisfactorily complete the story. Given the success of fine-tuning GPT-2 model, we plan to further investigate fine-tuning by tweaking certain hyper-parameters especially since it is unclear whether the additional training data and capacity of GPT-2 is sufficient to overcome the inefficiencies demonstrated by BERT.

6 Conclusion

We have seen that combining multiple neural network to create a holistic model that can generate text out of images provides highly relevant results. Each component of our model is performing its own individual task well, and they combine together effectively to perform the overall task of text generation from images.

While it is generally observed that combining multiple models to make a single prediction in machine learning algorithms generally leads to propagation of errors among the layers. However, the simple error minimization strategies we used were highly effective in keeping the resulting fictional story rooted to the context of the image.

We provide all of the code that we used to train the GPT-2 model, the final combined model and the processed short story database that we use for retrieval through our Github Repository: https://github.com/scarescrow/StoryReel

References

[1] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[2] Jiaao Chen, Jianshu Chen, and Zhou Yu. Incorporating structured commonsense knowledge in story completion. *CoRR*, abs/1811.00625, 2018.

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[4] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 329–340, New York, NY, USA, 2018. ACM.

[5] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. *Neural Architecture Search*, pages 63–77. Springer International Publishing, Cham, 2019.

[6] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *CoRR*, abs/1607.08221, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 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., 2012.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[10] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics.

[11] Nasrin Mostafazadeh, Lucy Vanderwende, Wen-tau Yih, Pushmeet Kohli, and James Allen. Story cloze evaluator: Vector space representation evaluation by predicting what happens next. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 24–29, Berlin, Germany, August 2016. Association for Computational Linguistics.

[12] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. *CoRR*, abs/1705.00393, 2017.

[13] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[15] Melissa Roemmele. Writing stories with help from recurrent neural networks. 2016.

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[17] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[18] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah Smith. The effect of different writing tasks on linguistic style: A case study of the roc story cloze task. pages 15–25, 01 2017.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[21] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[23] Zhihua Xie, Guodong Liu, and Zhijun Fang. Face recognition based on combination of human perception and local binary pattern. In Yanning Zhang, Zhi-Hua Zhou, Changshui Zhang, and Ying Li, editors, *Intelligent Science and Intelligent Data Engineering*, pages 365–373, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[24] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878, 2016.