DATA SCIENCE INTERVIEW PREPARATION

(30 Days of Interview Preparation)

Day26

Q1.What is DCGANs (Deep Convolutional Generative Adversarial Networks)?

Answer:

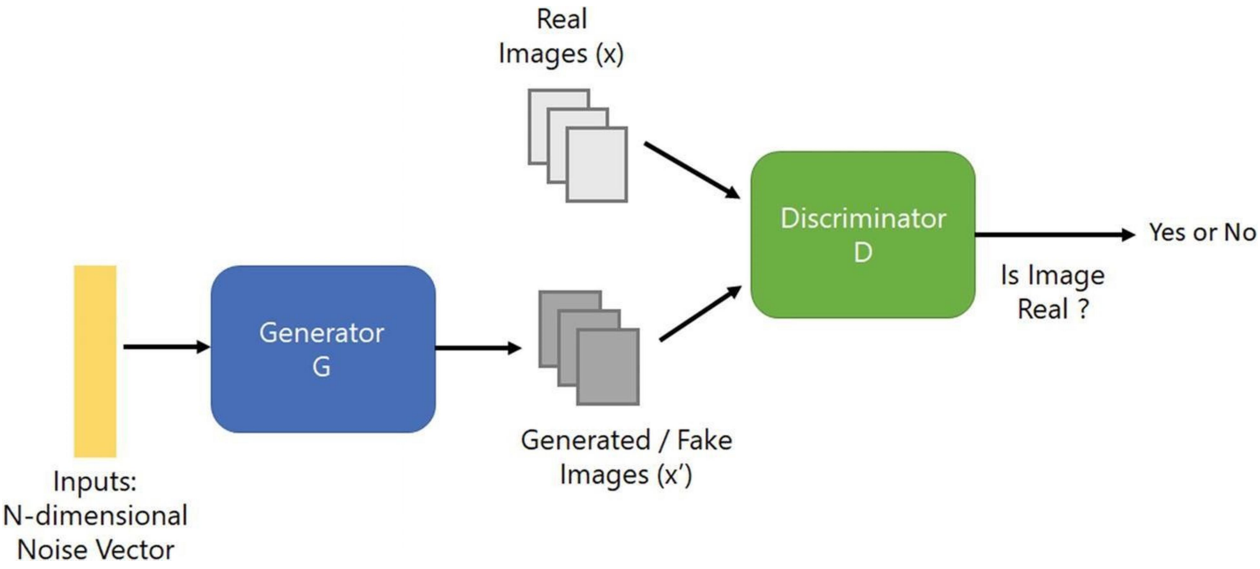
GANs stands for Generative adversarial networks, which is introduced by Ian Goodfellow in 2014. GANs is entirely new way of teaching computers how they do complex tasks through a generative process.

GANs have two components.

- A **Generator (**An artist**)** neural network.
- **A Discriminator (**An art critic**)** neural network.

Generator (An artist**)** generates an image. **Generator** does not know anything about the real images and learns by interacting with the **Discriminator**. The **Discriminator (**An art critic**)** determines whether an object is *"real"* and *"fake"* (usually represented by a value close to 1 or 0).

High-Level DCGAN Architecture Diagram



Original DCGAN architecture (Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*)* have four convolutional layers for **Discriminator** and "four fractionally-strided convolutions" layers for **Generator.**

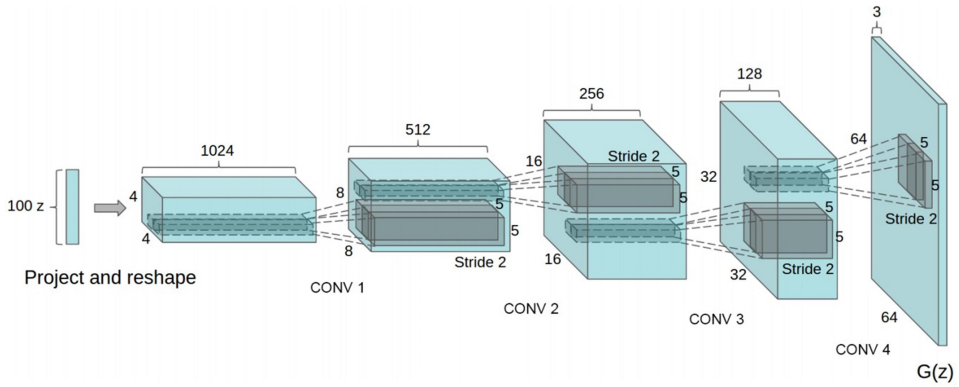


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

The Discriminator Network

The **Discriminator** is a *"art critic"* who tries to distinguish between "real" and "fake" images. This is a convolutional neural network(CNN) for image classification.

The **Discriminator** is 4 layers strided convolutions with batch normalization (except its input layer) and leaky ReLU activations. **Leaky ReLU** helps the gradients flow easier through the architecture.

The Generator Network

The **Generator** is *"*An artist*"* who tries to create an image that looks as "*real"* as possible, to fool **Discriminator**.

The **Generator** is four layers fractional-strided convolutions with batch normalization (except its input layer) and use **Hyperbolic Tangent (*tanh*)** activation in the final output layer and **Leaky ReLU** in rest of the layers.

Training of DCGANs

The following steps are repeated in training

- First **Generator** creates some new examples.
- And the **Discriminator** is trained using real data and generated data.

- After **Discriminator** has been trained, models are trained together.
- The **Discriminator**'s weights are frozen, but its gradients are used in **Generator** model so that **Generator** can update it's weights.

Q2. Explain EnAET.

Answer:

EnAET: Self-Trained Ensemble AutoEncoding Transformations for Semi-Supervised Learning

Deep neural network has shown its sweeping successes in learning from large-scale labeled datasets like ImageNet. However, such successes hinge on the availability of large amount of labeled examples that are expensive to collect. Moreover, deep neural networks usually have large number of parameters that are prone to over-fitting. Thus, we hope that semi-supervised learning can not only deal with limited labels but also alleviate the over-fitting problem by exploring unlabeled data. In this paper, we successfully prove that both goals can be achieved by training the semi-supervised model built upon self-supervised representations.

Semi-Supervised Learning (SSL) has been extensively studied due to its great potential for addressing the challenge with limited labels. Most state-of-the-art approaches can be divided into two categories. One is confident predictions, which improves a model's confidence by encouraging low entropy prediction on unlabeled data. The other category imposes consistency regularization by minimizing discrepancy among the predictions by different models. The two approaches employ reasonable objectives since good models should make confident predictions that are consistent with each other.

On the other hand, a good model should also recognize the object even if it is transformed in different ways. With deep networks, this is usually achieved by training a model with augmented labeled data. However, unsupervised data augmentation is preferred to explore effect of various transformations on unlabeled data. For this reason, we will show that self-supervised representations learned from auto-ending the ensemble of spatial and non-spatial transformations can play a key role in significantly enhancing semi-supervised models. To this end, we will present an Ensemble of Auto-Encoding Transformations (AETs) to self-train semi-supervised classifiers with various transformations by combining the advantages of both existing semi-supervised approaches and the newly developed self-supervised representations.
Our contributions are summarized as follows:

- We propose first method that employs ensemble of both spatial and non-spatial transformations from both labeled and unlabeled data in the self-supervised fashion to train a semi-supervised network.
- We apply an ensemble of AETs to learn robust features under various transformations, and improve the consistency of the predictions on transformed images by minimizing their KL divergence.
- We demonstrate EnAET outperforms the state-of-the-art models on all benchmark datasets in both semi-supervised and fully-supervised tasks.
- We show in the ablation study that exploring an ensemble of transformations plays a key role in achieving new record performances rather than simply applying AET as a regularize.

Q3. What is Data Embedding Learning?

Answer:

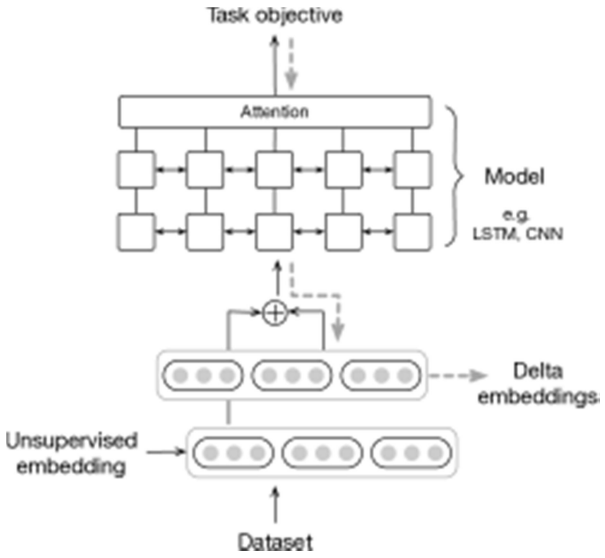
Unsupervised word embeddings have become basis for word representation in NLP tasks. Models such as skip-gram and Glove capture statistics of large corpus and have good properties that corresponds to semantics of word. However there are certain problems with unsupervised word embeddings, such as difficulty in modeling some fine-grained word semantics. For e.g., words in the same category but with different polarities are often confused because those words share common statistics in the corpus.

In supervised NLP(Natural Language Processing) tasks, these unsupervised word embeddings are often used in one of 2 ways: keeping fixed or using as initialization (fine-tuning). The decision is made based on amount of available training data in order to avoid overfitting. Nonetheless, underfitting with keeping fixed and certain degree of overfitting with fine-tuning is inevitable. Because this all are none optimization of word embeddings lacks control over the learning process, embeddings are not trained to an optimal point, which can result in suboptimal task performance.

In this paper, we propose delta embedding learning, the novel method that aims to address a above problems together: using regularization to find optimal fine-tuning of word embeddings. Better task performance can be reached with properly optimized embeddings. At the same time, regularized fine-tuning effectively combines semantics from supervised learning and unsupervised learning, which addresses some limitations in unsupervised embeddings and improves quality of embeddings.

Unlike retrofitting, which learns directly from lexical resources, our method provides the way to learn word semantics from supervised NLP(Natural Language Processing) tasks. Embeddings usually become task-specific and lose its generality when trained along with the model to maximize a task objective. Some approach tried to learn reusable embeddings from NLP(Natural Language Processing) tasks include multi-task learning, where one predict context words and external labels at the same time, and

specially designed gradient descent algorithms for fine-tuning .Our method learns reusable supervised embeddings by fine-tuning unsupervised embeddings on supervised task with a simple modification. The method also makes it possible to examine and interpret the learned semantics.



The aim of a method is to combine the benefits of unsupervised learning and supervised learning to learn better word embeddings. Unsupervised word embeddings like skip-gram, trained on large corpus (like Wikipedia), gives good-quality word representations. We use such embedding Wunsup as the starting point and learn a delta embedding wΔ on top of it:

$$\mathbf{w} = \mathbf{w}_{unsup} + \mathbf{w}_{\Delta}. \quad (1)$$

The unsupervised embedding Wunsup is fixed to preserve good properties of the embedding space and word semantics learned from large corpus. Delta embedding wΔ is used to capture discriminative word semantics from supervised NLP(Natural Language Processing) tasks and is trained together with model

for the supervised task. In order to learn useful word semantics rather than task-specific peculiarities that results from fitting (or overfitting) the specific task, we impose L21 loss, one kind of structured regularization on wΔ:

$$loss = loss_{task} + c\sum_{i=1}^{n}(\sum_{j=1}^{d} w_{\Delta ij}^2)^{\frac{1}{2}} \quad (2)$$

The regularization loss is added as extra term to loss of supervised task.

The effect of L21 loss on wΔ has straightforward interpretation: to minimize total moving distance of word vectors in embedding space while reaching optimal task performance. The L2part of a regularization keeps change of word vectors small, so that it does not lose its original semantics. The L1 part of regularization induces sparsity on delta embeddings, that only small number of words get non-zero delta embeddings, while majority of words are kept intact. The combined effect is selective fine-tuning with moderation: delta embedding capture only significant word semantics that is contained in the training data of a task while absent in the unsupervised embedding.

Q4. Do you have any idea about Rookie?

Answer:

Rookie: A unique approach for exploring news archives

News archives offer the rich historical record. But if the reader or the journalist wants to learn about new topic with a traditional search engine, they must enter query and begin reading or skimming old articles one-by-one, slowly piecing together intricate web of people, organizations, events, places, topics, concepts and social forces that make up "the news."

We propose Rookie, which began as attempt to build a useful tool for journalists. With Rookie, a user's query generates an interactive timeline, a list of important related subjects, and summary of matching articles—all displayed together as a collection of interactive linked views. Users click and drag along the timeline to select certain date ranges, automatically regenerating the summary and subject list at interactive speed. The cumulative effect: users can fluidly investigate complex news stories as they evolve across time. Quantitative user testing shows how this system helps users better understand complex topics from documents and finish a historical sensemaking task 37% faster than with a traditional interface. Qualitative studies with student journalists also validate the approach.

We built the final version of Rookie following eighteen months of iterative design and development in consultation with reporters and editors. Because the system aimed to help real-world journalists, the software which emerged from the design process is dramatically different from similar academic efforts. Specifically, Rookie was forced to cope with limitations in the speed, accuracy and interpretability of current natural language processing techniques. We think that understanding and designing around such limitations is vital to successfully using NLP in journalism applications; a topic which, to our knowledge, has not been explored in prior work at the intersection of two fields.

How it works?

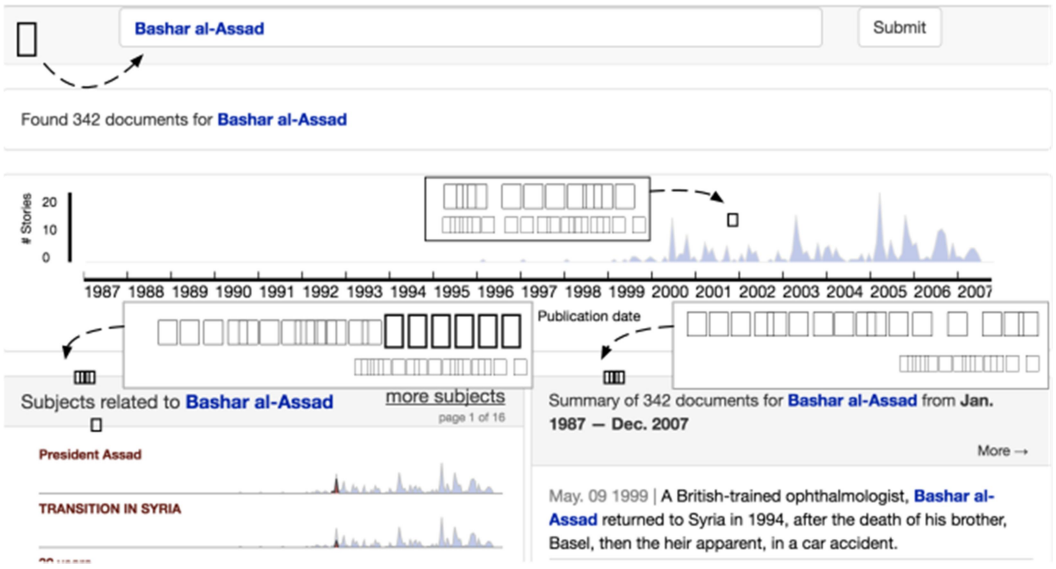
At any given time, Rookie's state is defined with the **user selection state**, the triple (**Q**,**F**,**T**) where:

- **Q** is the free text query string (e.g. “Bashar al-Assad”)
- **F** is related subject string (e.g. “30 years”) or is null
- **T** is time-span (e.g. Mar. 2000–Sep. 2000); by default, this is set to span of publication dates in the corpus.

Users first interact with Rookie by entering a query, **Q** into a search query bar using a web browser. For example, in below figure, a user seeking to understand the roots of the Syrian civil war has entered **Q** = “Bashar al-Assad”. In response, Rookie renders interactive time series visualization showing the frequency of matching documents from the corpus, a list of subjects in the matching documents, called subjects-sum and a textual summary of those documents, called sentence-sum.1In this example, the corpus is the collection of *New York Times* world news articles from 1987 to 2007 that contain the string “Syria”. All of the country-specific examples in this study are subsets of the same *New York Times* LDC corpus.

After entering **Q**, user might notice that “Bashar al-Assad” is mainly mentioned from 1999 onwards. To investigate, they might adjust time series slider to a spike in early mentions of Bashar al-Assad, **T** =Mar. 2000–Sep. 2000.

When user adjusts **T** to Mar. 2000–Sep. 2000, sentence-sum and subjects-sum change to reflect the new timespan below figure(c). subjects-sum now shows subjects like “TRANSITION IN SYRIA”,2Formatting from NYT section header style. “President Assad”, “eldest son” and “30 years” which are important to **Q** during **T**. (Bashar al-Assad’s father ruled for 30 years).



A) A user enters **Q** =“Bashar al-Assad” in order to learn more about the Syrian civil war.

At this point, user might explore further by investigating related subject, **F** ="President Assad"—clicking to select. sentence-sum now attempts to summarize the relationship between **Q**="Bashar al-Assad" and **F** ="President Assad" during **T** =Mar. 2000–Sep. 2000 figure(d). For instance, sentence-sum now shows the sentence: "Bashar al-Assad was born on Sept. 11, 1965, in Damascus, the third of President Assad's five children." If a user wants to understand this sentence in context, they can click sentence—which opens underlying document in the modal dialog.

F and **Q** are assigned red and blue colors throughout interface, allowing user to quickly scan for information. Bolding **Q** and **F** give additional clarity, and helps ensure that Rookie still work for colorblind users.

This example demonstrates how Rookie's visualization and summarization techniques work together to offer linked views of the underlying corpus. Linked views (a.k.a. multiple coordinated views) interfaces are common tools for structured information: each view displays the same selected data in a different dimension (e.g. a geographic map of a city which also shows a histogram of housing costs when a user selects a neighborhood). In Rookie's case, linked views display different levels of resolution. The time series visualization offers a **temporal view** of query-responsive documents, subjects-sum displays a medium-level **lexical view** of important subjects within the documents, and sentence-sum displays a low-level **text view** of parts of the underlying documents. The documents themselves, available by clicking extracted sentences, offer the most detailed level of zoom. Thus Rookie supports the commonly advised visualization pathway: "overview first, zoom and filter, and details on demand" (Shneiderman 1996).

Note that we use term **summarization** to mean selecting short text, or sequence of short texts, to represent a body of text. By this definition, both subjects-sum and sentence-sum are a form of summarization, as each offers a textual representation of the corpus—albeit at two different levels of resolution, phrases and sentences.

Q5.SECRET: Semantically Enhanced Classification of Real-world Tasks

Answer:

Significant progress has been made in NLP(natural language processing) and supervised machine learning (ML) algorithms over the past 2 decades. NLP successes include machine translation, speech or emotion or sentiment recognition, machine reading, and social media mining. Hence, NLP(Natural Language Processing) is beginning to become widely used in real-world applications that include either text or speech. Supervised Machine Learning(ML) algorithms excel at modeling the data-label relationship while maximizing performance and minimizing energy consumption and latency.

Supervised ML algorithms train on feature-label pairs to model the application of interest and predict labels. The label involves semantic information. use this information through vector representations of words to find novel class within the dataset. Karpathy and Fei-Fei generate figure captions based on the collective use of image datasets and word embeddings. Such studies indicate that data feature and semantic relationship correlate well. However, current supervised (Machine Learning)ML algorithms do not utilize such correlations in the decision-making (or prediction) process. Their decisions are based on the feature-label relationship, while neglecting significant information hidden in labels, i.e., meaning-based (semantic) relationships among label. Thus, they are not able to exploit synergies between feature and semantic space.

In this article, we show above synergies can be exploited to improve prediction performance of Machine Learning(ML) algorithms. Our method, called SECRET, combines vector representations of label in semantic space with available data in feature space within various operations (e.g., ML hyperparameter optimization and confidence score computation) to make final decisions (assign labels to the datapoints). Since SECRET does not target any particular Machine Learning(ML) algorithm or data structure, it is widely applicable.

The main contributions of this article are as follows:

- We introduce the dual-space Machine Learning(ML) decision process called SECRET. It combines new dimension (semantic space) with traditional (single-space) classifiers that operate in feature space. Thus, SECRET not only utilizes available data-label pairs, but also take advantage of meaning-based (semantic) relationships among labels to perform classification for the given real-world task.
- We demonstrate the general applicability of SECRET on various supervised Machine Learning(ML) algorithms and wide range of datasets for various real-world tasks.
- We demonstrate advantages of SECRET's new dimension (semantic space) through detailed comparisons with traditional Machine Learning(ML) approaches that have same processing and information (except semantic) resources.
- We compare the semantic space Machine Learning(ML) model with traditional approaches. We shed light on how SECRET builds semantic space component and its impact on overall classification performance.

Q6. Semantic bottleneck for computer vision tasks

Answer:

Image-to-text tasks have made tremendous progress since the advent of deep learning(DL) approaches. The work presented in this paper builds on these new types of image-to-text functions to evaluate capacity of textual representations to semantically and fully encode visual content of images for

demanding applications, in order to allow prediction function to host semantic bottleneck somewhere in its processing pipeline. The main objective of semantic bottleneck is to play role of *explanation* of the prediction process since it offers opportunity to examine meaningfully on what ground will further predictions be made, and potentially decide to reject them either using human common-sense knowledge and experience, or automatically through dedicated algorithms. Such the explainable semantic bottleneck instantiates good tradeoff between prediction accuracy and interpretability.

Reliably evaluating the quality of explanation is not straightforward. In this work, we propose to evaluate the explainability power of the semantic bottleneck by measuring its capacity to detect the failure of the prediction function, either through an automated detector as, or through human judgment. Our proposal to generate surrogate semantic representation is to associate the global and generic textual image description (caption) and visual quiz in the form of small list of questions and answers that are expected to refine contextually the generic caption. The production of this representation is adapted to vision task and learned from the annotated data.

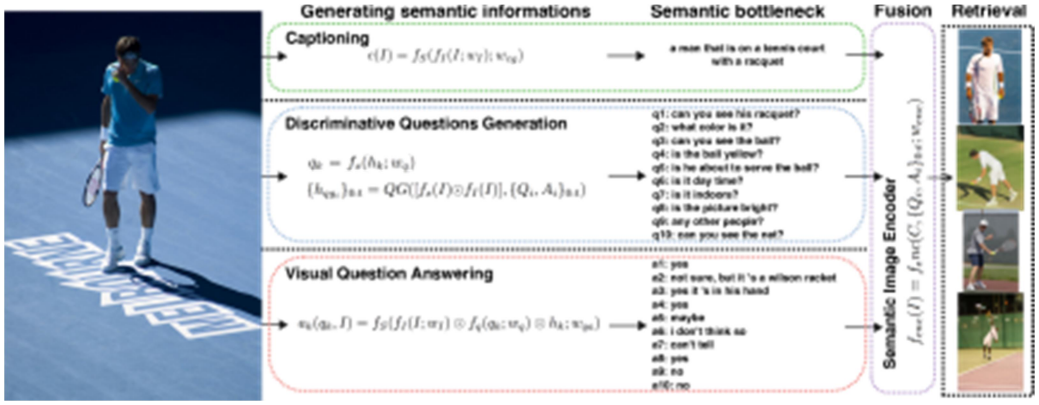


Figure : Semantic bottleneck approach: images are replaced by purely but rich textual representations, for tasks such as multi-label classification or image retrieval.

Q7. Gender Bias in Neural Natural Language Processing

Answer:

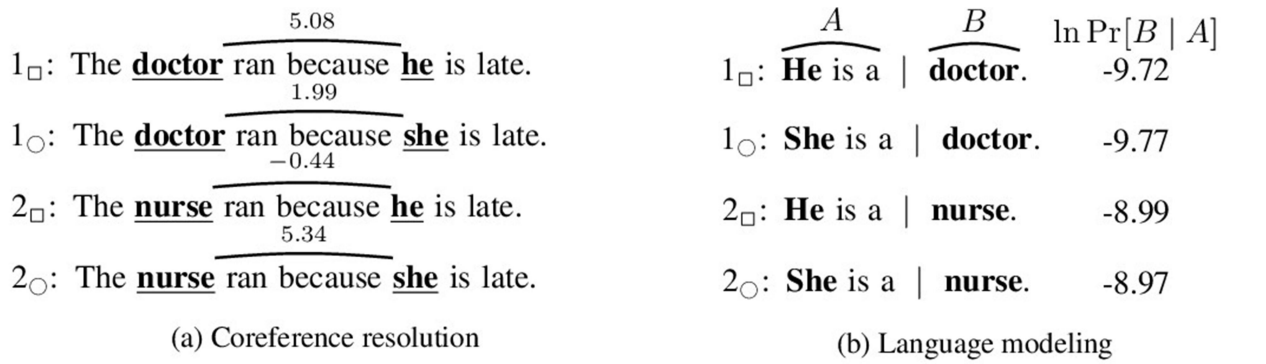
Natural language processing (NLP) with neural networks has grown in importance over last few years. They provide state-of-the-art(SOTA) models for tasks like coreference resolution, language modeling, and machine translation. However, since these models are trained on human language texts, natural question is whether they exhibit bias based on gender or other characteristics, and, if so, how should this bias be mitigated. This is a question that we address in this paper.

Prior work provides evidence of bias in autocomplete suggestions and differences in accuracy of speech recognition based on gender and dialect on popular online platforms. Word embeddings, initial pre-processors in many (Natural Language Processing)NLP tasks, embed words of a natural language into a vector space of limited dimension to use as their semantic representation. Observed that popular word embeddings including *word* exhibit gender bias mirroring stereotypical gender associations such as the eponymous "Man is to computer programmer as a Woman is to homemaker".

Yet the question of how to measure bias in general way for neural (Natural Language Processing)NLP tasks has not been studied. Our first contribution is general benchmark to quantify gender bias in variety of neural (Natural Language Processing)NLP tasks. Our definition of bias loosely follows idea of causal testing: matched pairs of individuals that differ in only the targeted concept (like gender) are evaluated by the model and the difference in outcomes (or scores) is interpreted as causal influence of the concept in scrutinized model. The definition is parametric in scoring function and the target concept. Natural scoring functions exist for number of neural natural language processing(NLP) tasks.

We instantiate definition for two important tasks—coreference resolution and language modeling. Coreference resolution is a task of finding words and expressions referring to the same entity in the natural language text. The goal of language modeling is to model distribution of word sequences. For neural coreference resolution models, we measure gender coreference score disparity between gender-neutral words and gendered words like the disparity between "doctor" and "he" relative to "doctor" and "she" pictured as edge weights in below Fig.. For language models, we measure disparities of emission log-likelihood of gender-neutral words conditioned on gendered sentence prefixes as is shown in below Fig. Our empirical evaluation with state-of-the-art(SOTA) neural coreference resolution and textbook RNN-based language models trained on benchmark datasets finds gender bias in these models.Note that these results have practical significance. Both coreference resolution and language modeling are core natural language processing(NLP) tasks in that they form the basis of many practical systems for information extraction, text generation, speech recognition and machine translation.

Next we turn our attention to mitigating the bias. Bolukbasi et al. (2016) introduced the technique for *debiasing* word embeddings which has been shown to mitigate unwanted associations in analogy tasks while preserving embedding's semantic properties. Given their spread use, a natural question is whether this technique is sufficient to eliminate bias from downstream tasks like coreference resolution and language modeling. As our 2nd contribution, we explore this question empirically. We find that while technique does reduce bias, residual bias is considerable. We further discover that debiasing models that make use of embeddings that are co-trained with their other parameters exhibit a significant drop in accuracy.



(a) Coreference resolution

(b) Language modeling

Figure 1: Examples of gender bias in coreference resolution and language modeling as measured

Q8. DSReg: Using Distant Supervision as a Regularizer

Answer:

Consider the following sentences in a text classification task, in which we want to identify sentences containing revenue values:

- *S1: The revenue of education sector is 1 million. (positive)*
- *S2: The revenue of education sector increased a lot. (hard-negative)*
- *S3: Education is a fundamental driver of global development. (easy-negative)*

S1 is a positive example since it contains precise value for the revenue, while both S2 and S3 are negative because they do not have the concrete information of revenue value. However, since S2 is highly similar to S1, it is hard for a binary classifier to make a correct prediction on S2. As another example, in reading comprehension tasks like NarrativeQA (Kočiskỳ et al., 2018) or MS-MARCO (Nguyen et al., 2016), truth answers are human-generated ones and might not have exact matches in the original corpus. A commonly adopted strategy is to first locate similar sequences from the original corpus using a ROUGE-L threshold and then treat these sequences as a positive training examples. Sequences that are semantically similar but right below this threshold will be treated as negative examples and thus inevitably introduce massive noise in training.

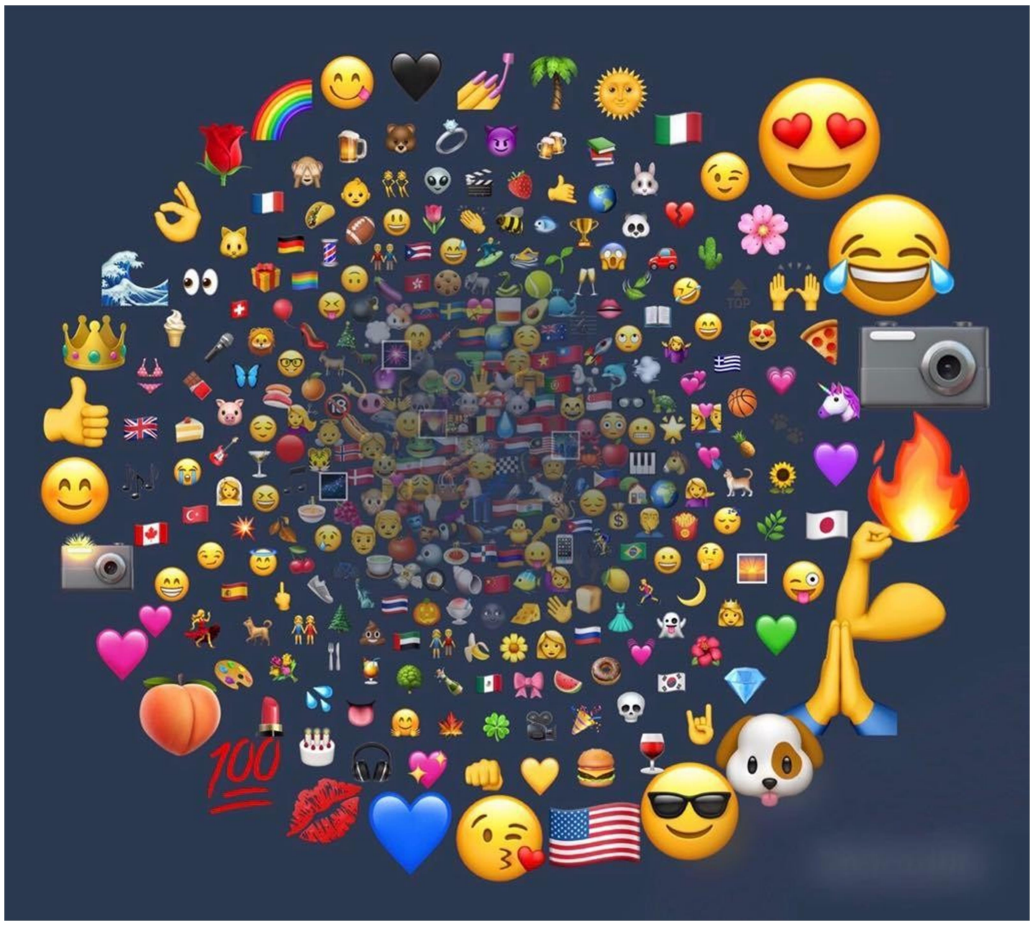
This problem is ubiquitous in a wide range of NLP tasks, i.e., when some of the negative examples are highly similar to the positive examples. We refer to these negative examples as *hard-negative examples* for the rest of this paper. Also, we refer to those negative examples that are not similar to the positive examples as *easy-negative examples*. If hard-negative examples significantly outnumber positive ones, features that they share in common will contribute significantly to the negative example category.

To tackle this issue, we propose using the idea of distant supervision to regulate the training. We first harvest hard-negative examples using distant supervision. This process can be done by the method as simple as using word overlapping metrics (e.g., ROUGE, BLEU or whether a sentence contains a certain keyword). With the harvested hard-negative examples, we transform the original binary classification task to a multi-task learning task, in which we jointly optimize the original target objective of distinguishing positive examples from negative examples along with an auxiliary objective of distinguishing hard-negative examples plus positive examples from easy-negative examples. For a neural network model, this goal can be easily achieved by using different softmax functions to readout the final-layer representations. In this way, both the difference and the similarity between positive examples and hard-negative examples can be captured by the model. It is worth noting that there are several key differences between this work and the mainstream works in distant supervision for relation extraction, at both the setup level and the model level. In traditional work on distant supervision for relation extraction, there is no training data initially and the distant supervision is used to get positive training data. In our case, we do have a labeled dataset, from which we retrieve hard-negative examples using the distant supervision.

Q9. What is Multimodal Emotion Classification?

Answer:

Emotion is any experience characterized by intense mental activity and certain degree of pleasure or displeasure. It primarily reflects all aspects of our daily lives, playing the vital role in our decision-making and relationships. In recent years, there have been growing interest in a development of technologies to recognize emotional states of individuals. Due to escalating use of social media, emotion-rich content is being generated at increasing rate, encouraging research on automatic emoji classification techniques. Social media posts are mainly composed of images and captions. Each of modalities has very distinct statistical properties and fusing these modalities helps us learn useful representations of data. Emotion recognition is the process that uses low-level signal cues to predict high-level emotion labels. With rapid increase in usage of emojis, researchers started using them as labels to train classification models.A survey conducted by secondary school students suggested that use of emoticons can help reinforce the meaning of the message. Researchers found that emoticons when used in conjunction with written message, can help to increase the "intensity" of its intended meaning.



Emojis are being used for visual depictions of human emotions. Emotions help us to determine interactions among human beings. The context of emotions specifically brings out complex and bizarre social communication. These social communications are identified as judgment of other person's mood based on his emoji usage (Rajhi, 2017). According to the study made by Rajhi et al. (Rajhi, 2017), the real-time use of emojis can detect the human emotions in different scenes, lighting conditions as well as angles in real-time. Studies have shown that emojis when embedded with text to express emotion make tone and tenor of message clearer. This further helps in reducing or eliminating the chances of misunderstanding, often associated with plain text messages. A recent study proved that co-occurrence allows users to express their sentiment more effectively.

Psychological studies conducted in the early '80s provide us strong evidence that human emotion is closely related to the visual content. Images can both express and affect people's emotions. Hence it is intriguing and important to understand how emotions are conveyed and how they are implied by visual content of images. With this as the reference, many computer scientists have been working to relate and learn different visual features from images to classify emotional intent. Convolutional Neural Networks (CNNs) have served as baselines for major Image processing tasks. These deep Convolutional Neural

Networks(CNNs) combine the high and low-level features and classify images in an end-to-end multi layer fashion.

Earlier most researchers working in field of social Natural Language Processing(NLP) have used either textual features or visual features, but there are hardly any instances where researchers have combined both these features. Recent works by Barbieri et al.'s, Illendula et al.'s, on multimodal emoji prediction and Apostolova et al. work on information extraction fusing visual and textual features have shown that combining both modalities helps in improving the accuracies. While a high percentage of social media posts are composed of both images and caption, researchers have not looked at multimodal aspect for emotion classification. Consider post in above Firgure where a user is sad and posts the image when a person close to him leaves him. The image represents the disturbed heart and has the textual description "sometimes tough if your love leaves you #sad #hurting" conveys a sad emotion. Similarly emoji used conveys emotion of being depressed. We hypothesize that all the modalities from the social media post including visual, textual, and emoji features, contribute to predicting emotion of the user. Consequently, we seek to learn importance of different modalities towards emotion prediction task.