DATA SCIENCE INTERVIEW PREPARATION (30 Days of Interview Preparation)

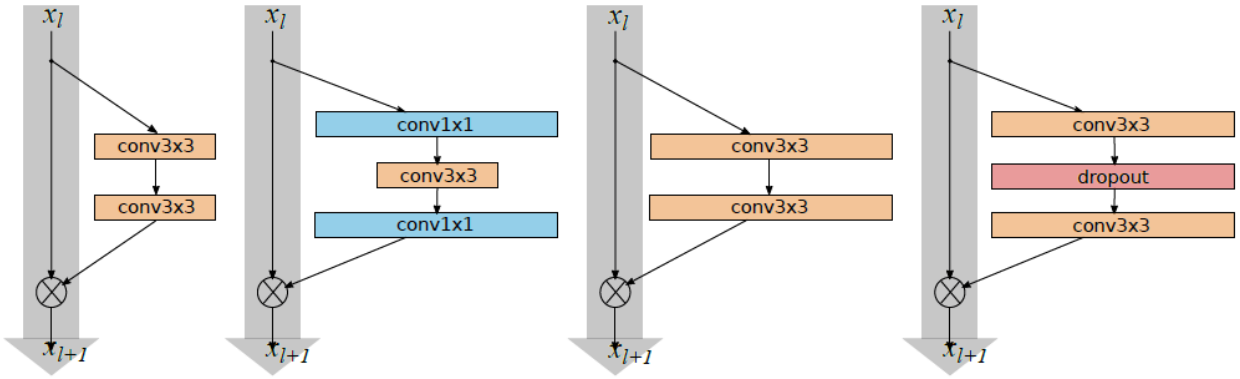
Day-25

Q1. What is WRN?

Answer:

WRN: It stands for Wide Residual Networks is presented. By widening Residual Network (ResNet), the network can be more superficial or shallow with same accuracy or improved accuracy. More external network means:

- the number of layers can be reduced.
- Training time can be shorter, as well.



(a) basic (b) bottleneck (c) basic-wide (d) wide-dropout

Problems on Residual Network (ResNet)

Circuit Complexity Theory

The authors of residual networks(ResNet) tried to make them as thin as possible in favor of increasing their depth and having less parameters and even introduced a «bottleneck» block, which makes ResNet blocks even thinner.

Diminishing Feature Reuse

However, As gradient flows through network, there is nothing to force it to go through residual block weights, and it can avoid learning anything during training, so there may be either only few blocks that learn useful representations or many blocks share very little information with a small contribution to the final goal. This problem was formulated as a diminishing feature reuse.

WRNs (Wide Residual Networks)

In WRNs, plenty of parameters are tested like the design of ResNet block, how deep (deepening factor l), and how extensive (widening factor k) within the ResNet block.

When *k*=1, it has the same width as the *ResNet*. While *k*>1, it is *k* time wider than *ResNet*.

WRN-*d*-*k*: means the WRN has a depth of *d* and with widening factor *k*.

- *Pre-Activation ResNet* is used in CIFAR-10, CIFAR-100, and SVHN datasets. Original *ResNet* is used in the ImageNet dataset.
- The significant difference is that *Pre-Activation ResNet* has the structure of performing batch norm and ReLU before convolution (i.e., BN-ReLU-Conv) while original *ResNet* has the structure of Conv-BN-ReLU. And *Pre-Activation ResNet* is generally better than the original one, but it has no visible improvement in ImageNet when layers are only around 100.

The design of the ResNet block

block type	depth	# params	time,s	CIFAR-10
$B(1,3,1)$	40	1.4M	85.8	6.06
$B(3,1)$	40	1.2M	67.5	5.78
$B(1,3)$	40	1.3M	72.2	6.42
$B(3,1,1)$	40	1.3M	82.2	5.86
$B(3,3)$	28	1.5M	67.5	5.73
$B(3,1,3)$	22	1.1M	59.9	5.78

-

- **B(3;3)**: Original «basic» block, in the above figure a.
- **B(3;1;3)**: With one extra (1×1) layer in between the two 3×3 layers
- **B(1;3;1)**: With the same dimensionality of all convolutions, bottleneck
- **B(1;3)**: The network has the alternating (1×1, 3×3) convolutions.
- **B(3;1)**: The network has the alternating(3×3, 1×1) convolutions.
- **B(3;1;1)**: This is Network in Network style block.

B(3;3) has the smallest error rate (5.73%).

Note: The Number of depths (layers) is different is to keep the number of parameters close to each other.

Q2.What is SIMCO: SIMilarity-based object Counting?

Answer:

Most approaches for counting similar objects in images assume a single object class; when is not, ad-hoc learning is necessary. None of them are genuinely agnostic and multi-class, i.e., able to capture repeated patterns of different types without any tuning. Counting approaches are based on density or regression estimation; here, we focus on counting by detection, so the counted objects are individually detected first.

Research on agnostic counting is vital in many fields. It serves for obvious question answering, where counting questions could be made on too-specific entities outside the semantic span of the available classes (e.g., "What is the most occurrent thing?" in below Fig.). In representation learning, unsupervised counting of visual primitives (i.e., visible "things") is crucial to obtain a rich image representation. Counting is a hot topic in cognitive robotics, where autonomous agents learn by separating sensory input into the finite number of classes (without a precise semantics), building the classification system that

counts on each of them.

Application-wise, agnostic counting may help the manual tagging of training images, providing a starting guess for the annotator on single- or multi-spectral images. Inpainting filters may benefit from a magic wand capturing repeated instances to remove.

Figure: SIMCO on obvious question answering: the most occurrent object? SIMCO finds 47 LEGO heads.

In this paper, we present the SIMCO (SIMilarity-based object COunting) approach, which is entirely agnostic, i.e., with no need for any *ad-hoc* class-specific fine-tuning, and multi-class, i.e., finding different types of repeated patterns. Two main ideas characterize SIMCO.

First, every object to be counted is considered as a specialization of a basic 2D shape: this is particularly true with many and small objects (see in above Fig: LEGO heads can be approximated as circles). SIMCO incorporates this idea building upon the novel Mask-RCNN-based classifier, fine-tuned just once on a novel synthetic shape dataset, *InShape*.

The second idea is that leveraging on the 2D shape approximation of objects; one can naturally perform unsupervised grouping of the detected objects (grouping circles with circles, etc.), discovering *different* types of repeated entities (without resorting to a particular set of classes). SIMCO realizes this with a head branch in the network architecture implementing triplet losses, which provides a 64-dim embedding that maps objects close if they share the same shape class plus some appearance attributes. Affinity propagation clustering finds groups over this embedding.

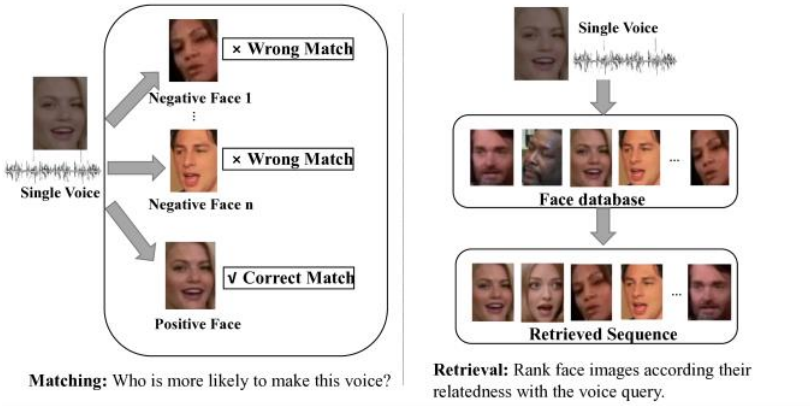
Q3. What is Voice-Face Cross-modal Matching and Retrieval?

Answer:

Studies in biology and neuroscience have shown that human's appearances are associated with their voices. Both the facial features and voice-controlling organs of individuals are affected by hormones and genetic information. Human beings can recognize this association. For example: when hearing from the phone call, we can guess the gender, the approximate age of the person on the other end of the line. When watching an unvoiced TV show. We can imagine an approximate voice by observing the face movement of the protagonist. With the recent advances of deep learning, face recognition models, and speaker recognition models have achieved exceptionally high precision. Can the associations between voices and faces be discovered algorithmically by machines? The research on this problem can benefit a lot of applications such as synchronizing video faces and talking sound, generating faces according to voices.

In recent years, much research attention has been paid on the voice-face cross-modal learning tasks, which have shown the feasibility of recognizing voice-face associations. This problem is generally formulated as a voice-face matching task and the voice-face retrieval task, as shown in Figure 1. Given a set of voice audios and faces, voice-face matching is to tell which look makes the voice when machine hearing voice audio. Voice-face retrieval is to present a sorted sequence of faces in the order of the

estimated match from a query of voice recording.



SVHF is the prior of voice-face cross-modal learning, which studies the performance of CNN-based deep network on this problem. The human's baseline for the voice-face matching task is also proposed in the paper. Both the "voice to face" and the "face to voice" matching tasks are studied in the Pins and Horiguchi's work, which exhibits similar performance on these two tasks. The curriculum learning schedule is introduced in Pins for hard negative mining. Various visualizations of the embedding vectors are presented to show the learned audio-visual associations in Kim's work. DIMNet learns the common representations for faces and voices by leveraging their relationship with some covariates such as gender and nationality. DIMNet obtains an accuracy of 84.12% on the 1:2 matching, which exceeds the human level.

Research on this problem is still in the early stage. Datasets used by previous research are always tiny, which can't evaluate the generalization ability of models sufficiently. Traditional test schemes based on random tuple mining tend to have low confidence. The benchmark for this problem needs to be established. This paper presents the voice-face cross-modal matching and retrieval framework, a dataset from Chinese speakers and a data collection tool. In the frame, cross-modal embeddings are learned with CNN-based networks, and triplet loss in a voice anchored metric space with L2-Norm constraint. An identity-based example sampling method is adopted to improve the model efficiency. The proposed framework achieves state-of-the-art performance on multiple tasks. For example, the result of 1:2 matching tested on 10 million triplets (thousands of people) reached 84.48%, which is also higher than DIMNet tested on 189 people. We have evaluated the various modules of the CNN-based framework and provided our recommendations. Even matching and retrieval based on the average of multiple voices and multiple faces are also attempted, which can further improve the performance. This task is the simplest way of analyzing video data. Large-scale datasets are used in this problem to ensure the generalization ability required in a real application. The cross-language transfer capability of the model is studied on the voice-face dataset of Chinese speakers we constructed. The series of performance

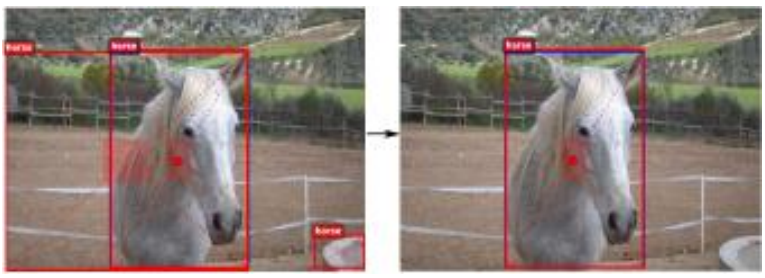
metrics are presented on the tasks by extensive experiments. The source code of the paper and the dataset collection tool will be published along with the article.

Q4. What is CenterNet: Object Detection with Keypoint Triplets?

Answer:

Object detection has been significantly improved and advanced with the help of deep learning, especially convolutional neural networks (CNNs). In the current era, one of the most popular flowcharts is anchor-based, which placed the set of rectangles with pre-defined sizes, and regressed them to the desired place with the help of the ground-truth objects. These approaches often need a large number of anchors to ensure the sufficiently high IoU (intersection over union) rate with the ground-truth objects, and the size and aspect ratio of each anchor box needs to be manually designed. Also, anchors are usually not aligned with the ground-truth boxes, which is not conducive to bounding box classification tasks.

To overcome the drawbacks of anchor based approaches, a keypoint-based object detection pipeline named CornerNet was proposed. It represented each object by a pair of corner key points, which bypassed the need for anchor boxes and achieved the state-of-the-art one-stage object detection accuracy. Nevertheless, the performance of CornerNet is still restricted by its relatively weak ability to refer to the global information of an object. That is to say since a pair of corners construct each object, the algorithm is sensitive to detect the boundary of objects, meanwhile not being aware of which pairs of critical points should be grouped into the objects. Consequently, as shown in Figure a, it often generates some incorrect bounding boxes, most of which could be easily filtered out with complementary information, *e.g.*, the aspect ratio.



To address this issue, we equip CornerNet with an ability to perceive the visual patterns within each proposed region, so that it can identify the correctness of each bounding box by itself. In this paper, we present the low-cost yet effective solution named **CenterNet**, which explores the central part of the proposal, *i.e.*, the region that is close to the geometric center, with one extra keypoint. Our intuition is that, if the predicted bounding box has a high IoU with the ground-truth box, then the probability that the center key point in its central region is predicted as the same class is high, and vice versa. Thus, during inference, after the proposal is generated as a pair of corner keypoints, we determine if the plan is indeed an object by checking if there is a crucial central point of the same class falling within its central

region. The idea, as shown in Figure a, is to use a triplet instead of a pair of key points to represent each object.

Accordingly, for better detecting the center keypoints and corners, we propose two strategies to enrich center and corner information, respectively. The first strategy is named as **center pooling**, which is used in the branch for predicting the center keypoints. Center pooling helps the center keypoints obtain more recognizable visual patterns within objects, which makes the central part of the proposal be better perceived. We achieve this by getting out the max summed response in both horizontal and vertical directions of the center key point on a feature map for predicting center keypoints. The second strategy is named **cascade corner pooling**, which equips the original corner pooling module with the ability to perceive internal information. We achieve this by getting out the max summed response in both boundary and inner directions of objects on a feature map for predicting corners. Empirically, we verify that such the two-directional pooling method is more stable, *i.e.,* being more robust to the feature-level noises, which contributes to the improvement of both precision and recall.

We evaluate the proposed CenterNet on the MS-COCO dataset, one of the most popular benchmarks for large scale object detection. CenterNet, with both center pooling and the cascade corner pooling incorporated, reports an AP of 47.0% on the test-dev set, which outperforms all existing one-stage detectors by the extensive margin. With an average inference time of 270ms using a 52-layer hourglass backbone and 340ms using a 104-layer hourglass backbone per image, CenterNet is quite efficient yet closely matches the state-of-the-art performance of the other two-stage detectors.

Q5. What is Task2Vec: Task Embedding for Meta-Learning?

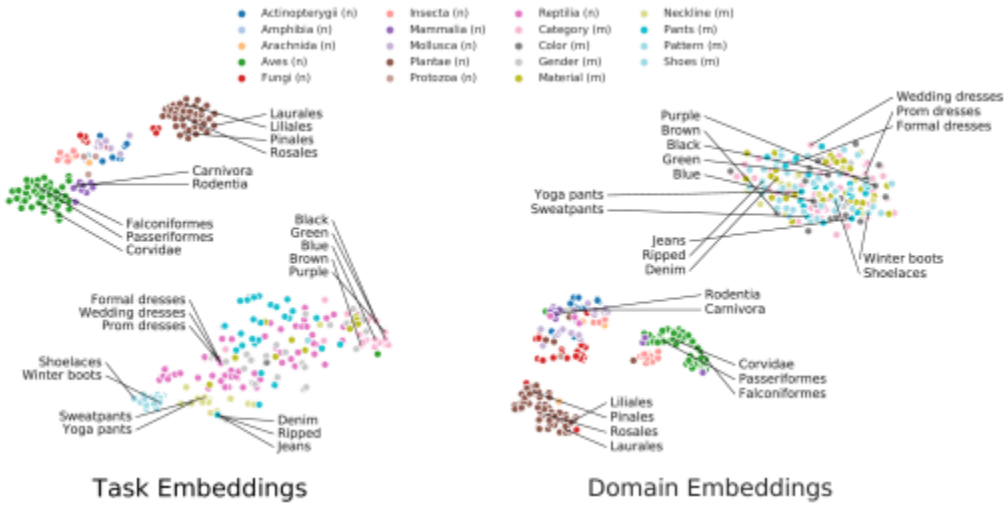
Answer:

The success of Deep Learning hinges in part on the fact that models learned for one task can be used on the other related tasks. Yet, no general framework exists to describe and learn relations between tasks. We introduce task2vec embedding, the technique to represent tasks as elements of the vector space is based on the Fisher Information Matrix. The norms of the embedding correlates with the complexity of the task, while the distance between embeddings captures the semantic similarities between tasks (Fig. 1). When other natural distances are available, such as taxonomical distance in the biological classification, we find that the embedding distance correlates positively with it (Fig. 2). Moreover, we introduce an asymmetric distance on tasks that correlates with the transferability between tasks.

Computation of the embedding leverages the duality between network parameters (weights) and outputs (activations) in a deep neural network (DNN): Just as the activations of a DNN trained on the complex visual recognition task are the rich representation of the input images, we show that the gradients of the weights relative to a task-specific loss are the rich representation of the task itself. Specifically, given a task defined by the dataset D={(xi,yi)}Ni=1 of labeled samples, we feed the data through a pre-trained

reference convolutional neural network which we call "*probe network*", and compute the diagonal Fisher Information Matrix (FIM) of the network filter parameters, to capture the structure of task. Since the architecture and weights of the probe network are fixed, the FIM provides the fixed-dimensional representation of the tasks. We show this embedding encodes the "difficulty" of the tasks, characteristics of the input domain, and features of the probe network are useful to solve it.

Our task embedding can be used to reason about the space of the tasks and solve meta-tasks. As a motivating example, we study the problem of selecting the best pre-trained feature extractor to solve a new task. This can be particularly valuable when there is insufficient data to train or fine-tune a generic model, and the transfer of knowledge is essential. task2vec depends solely on the task and ignores interactions with the model, which may, however, play an essential role. To address this, we learn about the joint task and model embedding, called model2vec, in such a way that models whose embeddings are close to a task exhibit excellent performance on the task. We use this to select an expert from the given collection, improving performance relative to fine-tuning a generic model trained on ImageNet and obtaining close to the ground-truth optimal selection.



Q6. What is GLMNet: Graph Learning-Matching Networks for Feature Matching?

Answer:

Many problems of interest in computer vision and pattern recognition area can be formulated as a problem of finding consistent correspondences between two sets of features, which are known as feature matching problem. Feature set that incorporates the pairwise constraint can be represented via an attribute

graph whose nodes represent the unary descriptors of feature points, and edges encode the pairwise relationships among different feature points. Based on this graph representation, feature matching can then be reformulated as a graph node matching problem.

Graph matching generally first operates with both node and edge affinities that encode similarities between the node and edge descriptors in two graphs. Then, it can be formulated mathematically as an Integral Quadratic Programming (IQP) problem with permutation constraint on related solutions to encode the one-to-one matching constraints. It is known to be NP-hard. Thus, many methods usually solve it approximately by relaxing the discrete permutation constraint and finding locally optimal solutions. Also, to obtain better node/edge affinities, learning methods have been investigated to determine the more optimal parameters in node/edge affinity computation. Recently, deep learning methods have also been developed for matching problems. The main benefit of deep learning matching methods is that they can conduct visual feature representation, node/edge affinity learning, and matching optimization together in an end-to-end manner. Zanfir et al. propose an end-to-end graph matching model, which makes it possible to learn all the parameters of the graph matching process. Wang et al. recently aim to explore graph convolutional networks (GCNs) for graph matching which conducts graph node embedding and matching simultaneously in a unified system.

Inspired by recent deep graph matching methods, in this paper, we propose a novel Graph Learning-Matching Network (GLMNet) for graph matching problems. Overall, the main contributions of this paper are three aspects.

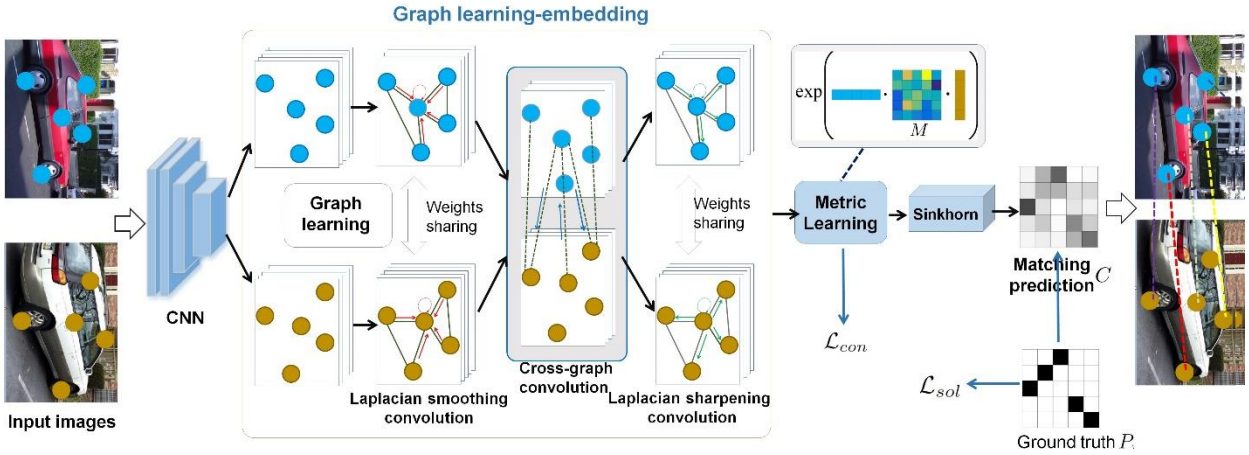
First, a critical aspect of (feature) graph matching is the construction of two matching graphs. Existing deep graph matching models generally use fixed structure graphs, such as k-NN, Delaunay graph, etc., which thus are not guaranteed to serve the parallel task best. To address this issue, we propose to adaptively learn a pair of optimal graphs for the matching task and integrate *graph learning* and *graph matching* simultaneously in a unified end-to-end network architecture.

Second, the existing GCN based graph matching model adopts the general smoothing based graph convolution operation for graph node embedding, which may encourage the feature embedding of each node becoming more similar to those of its neighboring nodes. This is desirable for graph node labeling or classification tasks, *but* undesirable for the matching task because extensive smoothing convolution may dilute the discriminatory information. To alleviate this effect, we propose to incorporate a Laplacian sharpening based graph convolution operation for graph node embedding and matching tasks. Laplacian sharpening process can be regarded as the counterpart of Laplacian smoothing which encourages the embedding of each node farther away from its neighbors.

Third, existing deep graph matching methods generally utilize a doubly stochastic normalization for the final matching prediction. This usually ignores the discrete one-to-one matching constraints in matching

optimization/prediction. To overcome this issue, we develop a novel constraint regularized loss to further incorporate the one-to-one matching constraints in matching prediction.

Experimental results, including ablation studies, demonstrate the effectiveness of our GLMNet and advantages of devised components, including graph learning-matching architecture, Laplacian sharpening convolution for discriminative embedding, and constraint regularized loss to encode one-to-one matching constraints.

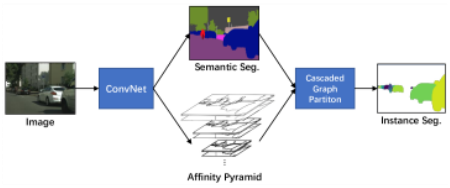


Q7. What is SSAP: Single-Shot Instance Segmentation With Affinity Pyramid?

Answer:

The rapid development of Convolutional networks has revolutionized various vision tasks, enabling us to move towards a more fine-grained understanding of images. Instead of classic bounding-box level object detection or class-level semantic segmentation, instance segmentation provides in-depth knowledge by segmenting all the objects and distinguish different object instances. Researchers are showing increasing interests in instance segmentation recently.

Current state-of-the-art solutions to this challenging problem can be classified into the *proposal-based* and *proposal-free* approaches. The proposal-based methods regard it as an extension to the classic object detection task. After localizing each object with a bounding box, the foreground mask is predicted within each bounding box proposal. However, the performances of the scheme based methods are highly limited by the quality of the bounding box predictions, and the two-stage pipeline also limits the speed of systems. By contrast, the proposal-free approach has the advantage of its efficient and straightforward design. This work also focuses on the proposal-free paradigm.



The proposal-free methods mostly start by producing instance-agnostic pixel-level semantic class labels, followed by clustering them into the different object instances with particularly designed instance-aware features. However, previous methods mainly treat the two sub-processes as the two separate stages and employ multiple modules, which is suboptimal. The mutual benefits between the two sub-tasks can be exploited, which will further improve the performance of the instance, segmentation. Moreover, employing multiple modules may result in additional computational costs for real-world applications.

To cope with the above issues, this work proposes a single-shot proposal-free instance segmentation method, which jointly learns the pixel-level semantic class segmentation and object instance differentiating in a unified model with a single backbone network, as shown in Fig. 1. Specifically, for distinguishing different object instances, an affinity pyramid is proposed, which can be jointly learned with the labeling of semantic classes. The pixel-pair affinity computes the probability that two pixels belong to the same instance. In this work, the short-range relationships for pixels close to each other are derived with dense small learning windows. Simultaneously, the long-range connections for pixels distant from each other are also required to group objects with large scales or nonadjacent parts. Instead of enlarging the windows, the multi-range relationships are decoupled, and long-range connections are sparsely derived from the instance maps with lower resolutions. After that, we propose learning the affinity pyramid at multiple scales along the hierarchy of a U-shape network, where the short-range and long-range affinities are effectively learned from the feature levels with the higher and lower resolutions respectively. Experiments in Table 3 show that the pixel-level semantic segmentation and the pixel-pair affinity pyramid based grouping are indeed mutually benefited from the proposed joint learning scheme. The overall instance of segmentation is thus further improved.

Then, to utilize the cues about global context reasoning, this work employs a graph partition method to derive instances from the learned affinities. Unlike previous time-consuming methods, the cascaded graph partition module is presented to incorporate the graph partition process with the hierarchical manner of the affinity pyramid and finally provides both acceleration and performance improvements. Concretely, with the learned pixel-pair affinity pyramid, the graph is constructed by regarding each pixel

as the node and transforming affinities into the edge scores. Graph partition is then employed from higher-level lower-resolution layers to the lower-level higher-resolution layers progressively. Instance segmentation predictions from the lower resolutions produce confident proposals, which significantly reduce node numbers at higher resolutions. Thus the whole process is accelerated.

Q8. What is TENER: Adapting Transformer Encoder for Name Entity Recognition?

Answer:

The named entity recognition (NER) is the task of finding the start and end of an entity in the sentence and assigning a class for this entity. NER has been widely studied in the field of natural language processing (NLP) because of its potential assistance in question generation Zhou et al. (2017), relation extraction Miwa and Bansal (2016), and coreference resolution Fragkou (2017). Since Collobert et al. (2011), various neural models have been introduced to avoid the hand-crafted features Huang et al. (2015); Ma and Hovy (2016); Lample et al.

NER is usually viewed as a sequence labeling task, the neural models typically contain three components: word embedding layer, context encoder layer, and decoder layer Huang et al. (2015); Ma and Hovy (2016); Lample et al. (2016); Chiu and Nichols (2016); Chen et al. Zhang et al. (2018); Gui et al. (2019b). The difference between various NER models mainly lies in the variance in these components.

Recurrent Neural Networks (RNNs) are widely employed in NLP tasks due to its sequential characteristic, which is aligned well with the language. Specifically, bidirectional extended short-term memory networks (BiLSTM) Hochreiter and Schmidhuber (1997) is one of the most widely used RNN structures. (Huang et al., 2015) was the first one to apply the BiLSTM and the Conditional Random Fields (CRF) Lafferty et al. (2001) to sequence the labeling tasks. Owing to BiLSTM's high power to learn the contextual representation of words, it has been adopted by the majority of the NER models as the encoder Ma and Hovy (2016); Lample et al. (2016); Zhang et al. (2018); Gui et al.

Recently, Transformer Vaswani et al. (2017) began to prevail in the various NLP tasks, like machine translation Vaswani et al. (2017), language modeling Radford et al. (2018), and pretraining models Devlin et al. (2018). The Transformer encoder adopts the fully-connected self-attention structure to model the long-range context, which is the weakness of RNNs. Moreover, the Transformer has better parallelism ability than RNNs. However, in the NER task, Transformer encoder has been reported to perform poorly Guo et al. (2019), our experiments also confirm this result. Therefore, it is intriguing to explore the reason why the Transformer does not work well in the NER task.

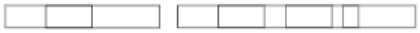


Figure 1: An example for NER. The relative direction is important in the NER task, because words before "Inc." are mostly to be an organization, words after "in" are more likely to be time or location. Besides, the relative distance between words is also important, since only continuous words can form an entity, the former "Louis Vuitton" can not form an entity with the "Inc.".

The first is that the sinusoidal position embedding used in the vanilla Transformer is relative distance sensitive and direction-agnostic, but this property will lose when used in the vanilla Transformer. However, both the direction and relative distance information are essential in the NER task. For example, words after "in" are more likely to be a location or time than words before it, and words before "Inc." is most likely to be of the entity type "ORG." Besides, an entity is a continuous span of words. Therefore, the awareness of relative distance might help the word better recognizes its neighbor. To endow the Transformer with the ability of directionality and relative distance awareness, we adopt direction-aware attention with the relative positional encoding Shaw et al. (2018); Huang et al. (2019); Dai et al. (2019). We propose a revised relative positional encoding that uses fewer parameters and performs better.

The second is an empirical finding. The attention distribution of the vanilla Transformer is scaled and smooth. But for NER, sparse attention is suitable since not all words are necessary to be attended. Given the current word, a few contextual words are enough to judge its label. The smooth attention could include some noisy information. Therefore, we abandon the scale factor of dot-production consideration and the use of un-scaled and sharp attention.

With the above improvements, we can significantly boost the performance of the Transformer encoder for NER.

Q9. What is Subword ELMo?

Answer:

Recently, pre-trained language representation has shown to be useful for improving many NLP tasks. Embeddings from Language Models is one of the most outstanding works, which uses the character-aware language model to augment word representation.

An essential challenge in training word-based language models is how to control the vocabulary size for better rare word representation. No matter how large the vocabulary is, unique words are always insufficiently trained. Besides, an extensive vocabulary takes too much time and computational resources for the model to converge. Whereas, if the dictionary is too small, the out-of-vocabulary (OOV) issue will harm the model performance slowly. To obtain effective word representation, Jozefowicz et al. (2016) introduce character-driven word embedding using the convolutional neural network (CNN).

However, potential insufficiency when modeling word from characters which hold little linguistic sense, especially, the morphological source. Only 86 roles are adopted in English writing, making the input too coarse for embedding learning. As we argue that for the better representation from a refined granularity, the word is too large, and character is too small, it is natural for us to consider subword units between character and the word levels.

Splitting the word into subwords and using them to augment the word representation may recover the latent syntactic or semantic information. For example, *uselessness* could be divided into the following subwords: Previous work usually considers linguistic knowledge-based methods to tokenize each word into the subwords (namely, morphemes). However, such treatment may encounter the three main inconveniences. First, the subwords from linguistic knowledge, typically including the morphological suffix, prefix, and stem, may not be suitable for the targeted NLP task Banerjee and Bhattacharyya or mislead the representation of some words, like the meaning of *understanding* cannot be formed by *under* and *stand*. Second, linguistic knowledge, including related annotated lexicons or corpora, may not even be available for the specific low-resource language. Due to these limitations, we focus on the computationally motivated subword tokenization approaches in this work.

In this paper, we propose Embedding from Subword-aware Language Models (ESuLMo), which takes subword as input to augment word representation and release a sizeable pre-trained language model research communities. Evaluations show that the pre-trained language models of ESuLMo outperform all RNN-based language models, including ELMo, in terms of PPL and ESuLMo beats state-of-the-art results in three of four downstream NLP tasks.