Generating referring expressions

Group2

The problem: how to visually ground text or map what is inside an image to words. Image contextual description could be automatically generated if the model understands the image semantics and generates text based on this understanding. Both papers [1,2] try to generate a textual description for an image automatically. The first paper [1] used the classical machine learning (ML) approach where an algorithm learns to map triplet of image elements to triplet of image description. Then, the algorithm uses this text triplet to generate the caption. Paper [2] utilized the encoder-decoder deep learning (DL) architecture to generate a caption for images automatically.

In Paper [1], the image is represented in a triplet of objects associated with attributes, actions/poses, and spatial relationships. The algorithm maps the objects to nouns, actions/poses to verbs, objects' attributes to nouns or adjectives, and spatial relations to prepositions. The algorithms, then, go into several engineered steps to generate the text using nouns as the cornerstones. For instance, to name but a few, the algorithm cluster the nouns —that represent the objects— into different groups if they are more than three. For each group, nouns are ordered by greedily maximizing the probability of a noun being in a specific location in the sentence. After that, detected adjectives are grouped into classes, called attributes, based on the similarity between the adjective and the attributes learned from the dataset. Then, by utilizing the learned adjective-noun co-occurrence, the algorithm selects the most likely word for each attribute; this choice of one-to-one matching is based on the fact that each attribute rarely appears more than once in the textual image description.

As we can see from the above discussion, the designer had to formalize the problem so that the algorithm can solve the problem. The algorithm consists of eight handcrafted steps[1]; for instance, the decision to cluster nouns is based on the observation that there are no more than three nouns (object) in one sentence. Such handcrafting operation is expensive and complicated, and it gets more and more complicated if the problem is complex. **In the second paper[2]**, instead of manually selecting and designing the feature, the deep network automatically learns them by showing it examples of the desired input-output behavior. The system is based on an encoder-decoder architecture with an attention mechanism. The encoder tries to extract the image features and encode them, and the decoder decodes these features into text. The spatial attention module tries to learn which region in the image is related to the generated word. The sentinel attention module tries to know whether the generated word needs to be visually grounded. Using of spatial attention module only will

The two papers have a common design feature; both try to understand the image semantics by extracting the important parts in the image that correlate with the text description. This correlation is done in paper [1] by mapping the image-triplet —predefined important aspects of image semantics that need to be described— to text-triplet mapping. While in article [2], this correlation is done by utilizing the spatial and the fallback sentinel attention modules.

According to [3], DL methods can generate contextually and semantically image captions richer than the classical ML method. This conclusion drags us to the discussion of **generative linguistics vs. neural network**. In [4], the author argued that DL provides a dramatic error reduction (25%-50%) only on true signal processing tasks; vision and speech recognition. On the other hand, this dramatic error reduction is not achievable by DL on higher-level language processing. We do not accurately understand what is meant by higher-level language processing and whether language generation or captioning image generation falls within the higher-level language processing. We realize —to the far of our understanding— that the conclusions/assumptions introduced by papers [3] and [4] are contradicting. It worth mentioning that the evaluation methods used in paper [1] and paper [2] are not comparable. However, we agreed that using traditional ML techniques or generating text based on synthetic constraints is more complicated than using DL techniques. Even though designing a DL system is not trivial as there are wide varieties of design parameters that need to be taken care of, like choosing the suitable architecture, tuning the hyperparameters, etc. The system design will get even relatively more complex, using handcrafted methods, if we need to design an agnostic language system.

As we discussed before, the most advantageous property that a deep neural network has is its ability to automatically learn features by showing it examples of the desired input-output behavior. Also, deep neural networks can **generalize concepts** to some extent by drawing complex non-linear borders between the data it learns from. To achieve such generalization capability, neural networks need a huge dataset and a strong computation capability. However, such capability comes with a negative side, which is a **lack of transparency**. In the training phase, the neural network determines its functionality by adapting its weights to the training dataset. These weights are big matrices of numbers that are not explainable. In paper [2 Figure 4], the authors try to give a qualitative evaluation by visualizing the generated caption and the visual grounding probability. There are multiple wrong generated captions; it is hard to explain why the DL system specifically chooses these (wrong) words. A DL system's behavior can not be anticipated, unlike the classical ML system that operates within a predefined formal framework. This issue raises a question: how the designer or the user can validate that the DL system meets the requirements and the specifications and can verify that it fulfills its intended purpose. The unclear relationship between the outputs and the inputs makes us question how failure assessment can be done on a DL system.

Both papers solve a problem by grounding individual words; however, we think humans do not correlate what is inside images with individual words. Also, generating a sentence or two sentences that provide a surface description of an image may be suitable for some content generation applications. However, we may need to have a system that provides a more emotionally engaging or informative or analytical image/scene description to be used in museums or hospitals or used in medical applications or used as assistance for visually impaired users.

Reference:

1. Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., & Daumé, H. (2012). Midge: Generating image descriptions from computer vision detections. *EACL 2012 - 13th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings*, 747–756.

2. Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 3242–3250. https://doi.org/10.1109/CVPR.2017.345
3. Zakir Hossain, M. D., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, *51*(6). https://doi.org/10.1145/3295748
4. Manning, C. D. (2015). Last Words: Computational Linguistics and Deep Learning. *Computational Linguistics*, *41* (4), 701-707. https://doi.org/doi:10.1162/COLI_a_00239