



Bidirectional Transformer for Android Based Image Icon Text Generation

Chuyi Yu^{1*}, Ying Ma², and Jianmin Li³

^{1,2,3} *Computer and Information Engineering, Xiamen University of Technology, No.600 Ligong Road, Jimei District, Xiamen, 361024, Fujian Province, China*

**Corresponding author. Email: yucy_2661@163.com*

Abstract:

The image caption task is an important manifestation of the fusion of computer vision and natural language processing development in deep learning. The Image Caption task, which is an advanced kind of image comprehension, can effectively grasp image information and produce accurate and concise natural language descriptions to users. It has gotten a lot of attention in the subject of art intelligence, and it has a lot of uses in the field of assisting visually impaired guides and human-computer interaction. This research primarily presents a deep learning-based solution for completing the natural language generation task based on images and symbols in Android. The encoder-decoder framework is used as the core structure to help visually impaired persons interact with mobile phones.

Keywords: *Image Caption, Natural Language Processing, Encoder-Decoder Framework*

1 INTRODUCTION

Mobile phones have become an indispensable element of modern life, allowing users to access services such as business, communications, healthcare, and transportation via a broad range of apps. Poor mobile app accessibility risks losing not only disabled users, but also a big aging population with varied degrees of handicap. According to World Health Organization figures, "Globally, around 39 million individuals are blind, and 246 million have impaired vision" [1]. These users' requirements must also be considered. As a result, mobile accessibility is critical, and well-known platforms like IOS and Android have auxiliary tools like Google TalkBack on Android and VoiceOver on IOS. When the user places their finger on the icon or text on the smartphone screen, screen readers read the information aloud. However, in order to help screen readers to comprehend the meaning of the screen, adequate text alternatives for the image content must be provided in advance. In fact, vast majority of images in Android applications currently lack labeling information [10], and manual labeling is time-consuming and labor-intensive. How to automatically generate natural language descriptions of images through computers is a challenging task in the field of artificial intelligence.

Image captioning's purpose is to automatically generate a natural language description of a given image, and it is a potential study field for deep learning's combination of image recognition and natural language processing. Xuedong Huang, a technical researcher at Microsoft Corporation, pointed out that image captioning is one of the core functions of computer vision, which can realize a wide range of services. Now image captioning has been widely used. Microsoft created a "Seeing AI" program in 2017 that uses the phone camera to describe items for individuals with vision issues, allowing them to see the world through their smartphone. Google has also released a tool that can generate textual descriptions of photographs, helping blind or visually impaired persons to comprehend the image or scene. Increasingly work focuses on opening new horizons for blind users. A recent study that received the ACM SIGSOFT Distinguished Paper Award made a big impression. Chen devised a method called LabelDroid that use deep learning to make it easier for visually impaired persons to use cellphones [3]. This paper is a research based on these works. We present BiLabel, a method that uses deep learning to annotate labels based on image icons in Android applications.

2 RELATED WORKS

2.1 Accessibility test

Accessibility testing [4] is commonly used in software testing to evaluate apps from the perspective of individuals with impairments and to guarantee that developers can create applications that are accessible to all users. The World Wide Web Consortium (W3C) publishes several implementation guidelines and technical standards for accessibility testing for developers to learn and follow. This implies that mobile app usability testing research is vital for acquiring a unique viewpoint on the most prevalent user concerns and is a crucial component of the user experience tool. However, the widespread use of mobile devices and the increasing rise of mobile devices make programmers' jobs more difficult. Therefore, there are many ways to ensure the accessibility of the program. We can choose either manual testing or automated tools to perform accessibility testing. Accessibility testing can be done manually or with the use of automated technologies. Appium is now the most popular mobile automated testing solution on the market, as it can handle automated testing of Android and iOS simulators as well as actual computers at the same time, and it can be written in nearly any programming language. Espresso, Google's open source automated testing framework, is smaller in scope, has a more precise API, and is easier to learn. Programmers can select appropriate testing tools based on their requirements. In addition to testing tools, we can also use manual testing. Manual testing can compensate for issues that testing technologies are unable to uncover. Google suggests using TalkBack and Toggle Access to test applications at the very least. However, despite the abundance of accessibility tools, developers still require a strong awareness of accessibility testing, therefore improved ways to lessen the load on programmers are required.

2.2 Image caption

Image processing, image identification, natural language processing, text creation, and other technologies have expanded significantly in recent years, with the strengthening of machine learning and deep learning research, and have become research hotspots. In computer vision applications such as image classification and object recognition, convolutional neural networks (CNN) have obtained exceptional application results. In natural language processing, recurrent neural networks (RNN) are also significant. Furthermore, inspired by the encoder-decoder structure in deep learning machine translation [11], encoding image features first and then generating sentences have become two fixed components of deep learning approaches based on the encoder-decoder structure. In general, any convolutional neural network architecture may be used for encoding, and the

majority of existing work either uses pre-trained CNN for feature extraction or Fast RCNN to identify objects first and then extract features. Many studies have been committed to improving encoders in order to extract more relevant information from images. For example, some studies add semantic information to visual input [5], or substitute CNN with an object detection module [14]. The notion of non-convolution has recently received a lot of attention. Liu developed and got good results using a totally convolution free model to accomplish the task of image annotation [8]. The decoder uses the already encoded image to create captions word for word. The anticipated words are then combined to form the following word. The attention mechanism is one of the most significant advances in the encoder-decoder structure. Object regions, attributes, and interactions between objects have all been thoroughly researched over the years, and the development of attention and early fusion approaches like BERT has made both phases greatly developed.

3 APPROACH

Given an image-based icon, our goal is to automatically predict the natural language description of the image. In this section, we introduce our model framework and implementation details.

3.1 Model details

For feature extraction, we utilize a non-convolutional method, slicing the image into N patches of the same size, flattening each patch into a vector, then mapping the vector to the feature space using a linear embedding layer to obtain the feature input of the image. We employ a transformer [12] based on encoder-decoder architecture as our basis model due to Transformer's excellent performance on machine translation tasks and its ability to be efficiently parallelized. Our encoder is made up of N identical layer stacks. Each layer is divided into two sublayers: a multi-head self-attention layer and a positionally completely linked feedforward neural network. The visual feature encoding is learned via the multi-head self-attention layer's scaled dot product attention. The following is the specific procedure:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

$$MHA(Q, K, V) = Concat(H_1, \dots, H_n)W^O \quad (2)$$

where Q, K, V denotes query, key, and value, and these vectors are formed by multiplying the image embedding by the three weight matrices created during our training phase. Finally, multiple attention results are merged together. The FFN layer is in charge of extracting additional features from the output of the multi-head self-

attention layer. The decoder, like the encoder, is made up of three sub-layer structures that are layered N times. To avoid seeing future information while forecasting, our decoding output for a sequence should only depend on the output before time t , not the output after time t . To hide the information after time t , we introduced a masking mechanism to the first self-attention sublayer. The second self-attention sublayer performs a multi-head attention operation on the output of the encoder and the output of the previous layer of the decoder to learn the relationship between visual features and ground-truth. Each sub-layer of the encoder and decoder is subjected to residual connections and layer normalization.

$$\text{sublayer} = \text{LayerNorm}(x + (\text{SubLayer}(x))) \quad (3)$$

Finally, the prediction result is obtained by passing the output of the top feedforward neural network through the softmax layer, and the decoder creates the output sequence one character at a time. In addition, for the decoder, we created a language model with the same network structure that decodes captions in both forward and reverse orientations, called forward decoder and reverse decoder, respectively. During training, the encoder's image features are fed into two decoders at the same time, the caption is predicted using bidirectional language, and the network is updated by the sum of the two cross entropies. Figure 1 depicts the specific structure of our model.

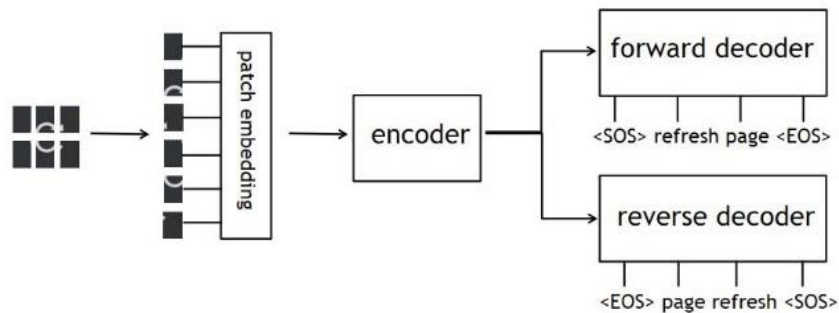


Figure 1: The overall architecture of our model BiLabel

3.2 Implementation details

The LabelDroid dataset [3], which contains 19,233 image-based buttons and content descriptions, is used for training. 15595, 1759, and 1879 are the train, validation, and test sets, respectively. We conducted necessary data augmentation processing on the photos throughout the training procedure. We crop the image at random and modify the brightness, contrast, and saturation at random. The size of our model is 512, the number of attention heads is 8, and the feedforward layer size is 2048. To compute the cross-entropy, we update the network with the model output and ground-truth during training and utilize the ADAM [6] optimizer. During training we update the network with the model output and ground-truth to compute the cross-entropy, and use the ADAM optimizer for training. We didn't use weight decay for residual connections and layer normalization in the transformer. Our model is implemented with PyTorch.

4 EXPERIMENT

4.1 Evaluation metrics

We evaluate our model using the following image captioning evaluation measures. BLEU [9] is a popular automated assessment metric in machine translation research, which compares the machine-generated translation to the reference sentence and calculates the similarity. We set n to 1, 2, 3, and 4 to count the BLEU values, denoted as BLEU_1, BLEU_2, BLEU_3, and BLEU_4. METEOR expresses more relevance at the sentence level and is a popular assessment statistic in machine translation. METEOR [2] aligns test and reference sentences utilizing word exact match, synonym match, and WordNet-based alignment. The alignment results are then used to calculate the similarity score between the test and reference sentences. ROUGE [7] is a measure that is based on recall. In this study, we use ROUGE-L, focusing on the longest common subsequence between the test and reference sentences. The two sentences are more similar if their shared subsequences are longer. CIDEr [13] is an image captioning assessment measure. The authors contend that while earlier assessment criteria had good relationships with people, their likeness to humans cannot be assessed.

It treats each phrase as a document, then computes the cosine angle of the TF-IDF vector, calculates the similarity between the candidate and reference sentences, and then averages the results.

4.2 Experimental results

We compare our method to two model structures that are extensively employed in image captioning. The first is to utilize CNN to extract features and then use the transformer as the encoder-decoder structure, while the second is to combine the image block and the transformer

as the encoding and decoding framework. Our values are shown as percentages in Table 1, which reflects our model's overall performance. The table 1 shows that the performance of the two infrastructures is comparable. The Vit+Transformer variant is marginally superior, although both are significantly less expensive than BiLabel. Our model have an exact match rate of 48.3%, which is 5% higher than the Vit+Transformer model. Bleu_1 and Bleu_2 had average accuracy of 53.1% and 51.3%, respectively. CIDEr reaches 24.7%, whereas Bleu_3 improves by 9%. It can be seen that our BiLabel has a significant effect.





Table 1: Compare with two baseline on LabelDroid dataset.

	Exact match	Bleu_1	Bleu_2	Bleu_3	Bleu_4	Meteor	Rouge_L	CIDEr
CNN+Transformer	41.1	43.7	42.0	30.9	18.3	26.4	44.1	21.4
Vit+Transformer	43.5	48.9	44.6	30.2	22.7	45.9	52.4	23.4
BiLabel	48.3	53.1	51.3	39.9	27.7	31.0	52.5	24.7

In addition, we also perform qualitative result analysis on our model. Table 2 illustrates several qualitative instances from the LabelDroid dataset. Because of the commonality of the symbols, our two basic models were able to keep some accuracy in the first two cases. In the third example, this sort of icon is frequently misinterpreted due to its aesthetic resemblance to the "forward" icon. However, our approach BiLabel

captures more features in images than baseline methods and can predict image labels correctly. In the fourth example, all model predictions are not identical to ground-truth, but our model's prediction outputs "add list" and "add information" may be substituted equivalently, demonstrating that our model still has a high level of innovation.

Table 2: Qualitative result analysis on our model.

	CNN+ Transformer	Vit+ Transformer	BiLabel	Ground-truth
	more option	<unk> option	more option	more option
	clear	delete	delete	delete
	navigate up	go forward	backward	backward
	<unk>	add <unk>	add list	add information

5 CONCLUSION

In this paper, we extend the standard transformer model with a reverse decoder that can run in parallel with the forward decoding process, implicitly utilizing bidirectional context information. On the LabelDroid dataset, we undertake research and validation, proving the importance of our bidirectional decoder. We also achieve good results by combining convolution-free feature extraction with our model.

REFERENCES

- [1] Alshayban A, Ahmed I, Malek S. Accessibility issues in android apps: state of affairs, sentiments, and ways forward[C]//2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE). IEEE, 2020: 1323-1334.
- [2] Banerjee S, Lavie A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments[C]//Proceedings of the acl

- workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. 2005: 65-72.
- [3] Chen J, Chen C, Xing Z, et al. Unblind your apps: Predicting natural-language labels for mobile gui components by deep learning[C]//2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE). IEEE, 2020: 322-334.
- [4] Eler M M, Rojas J M, Ge Y, et al. Automated accessibility testing of mobile apps[C]//2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST). IEEE, 2018: 116-126.
- [5] Fu K, Jin J, Cui R, et al. Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(12): 2321-2334.
- [6] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [7] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Text summarization branches out. 2004: 74-81.
- [8] Liu W, Chen S, Guo L, et al. Cptr: Full transformer network for image captioning[J]. arXiv preprint arXiv:2101.10804, 2021.
- [9] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002: 311-318.
- [10] Ross A S, Zhang X, Fogarty J, et al. Examining image-based button labeling for accessibility in Android apps through large-scale analysis[C]//Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility. 2018: 119-130.
- [11] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.
- [12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [13] Vedantam R, Lawrence Zitnick C, Parikh D. Cider: Consensus-based image description evaluation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4566-4575.
- [14] You Q, Jin H, Wang Z, et al. Image captioning with semantic attention[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 4651-4659.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

