Evaluating and interpreting caption prediction for histopathology images

Renyu Zhang[1], Christopher Weber[2], Robert Grossman[1] and Aly Khan[2]
1. Dept. of Computer Science, 2. Dept. of Pathology, The University of Chicago

Abstract

The automatic generation of captions from medical images can provide for an efficient way to annotate histopathology images with natural language descriptions. Such large-scale annotation of medical images may help facilitate image retrieval tasks and standardize clinical ontologies. In this work, we focus on developing and methodically evaluating a new caption generation framework for histopathology whole-slide images. We introduce PathCap, a deep learning multi-scale framework, to predict captions from histopathology images using multi-scale views of whole-slide images. We demonstrate that our framework outperforms a standard baseline caption model on a diverse set of human tissues and provides interpretable contextual cues for understanding predicted captions. Finally, we draw attention to a novel dataset of histopathology images with captions from the Genotype-Tissue Expression (GTEx) project, providing a valuable dataset for the machine learning and healthcare community to benchmark future caption prediction and interpretation methods.

Example

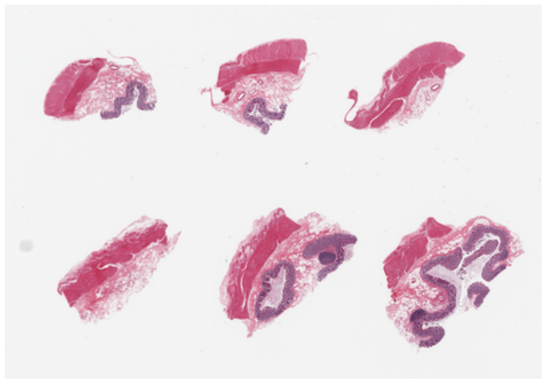


Figure 1: ``6 pieces; 4 pieces have full thickness elements with well preserved mucosa; 2 have no mucosa (in this section).'' (Example slide and caption from GTEx https://www.gtexportal.org/ sample: GTEX-131XE-0826).

Architecture

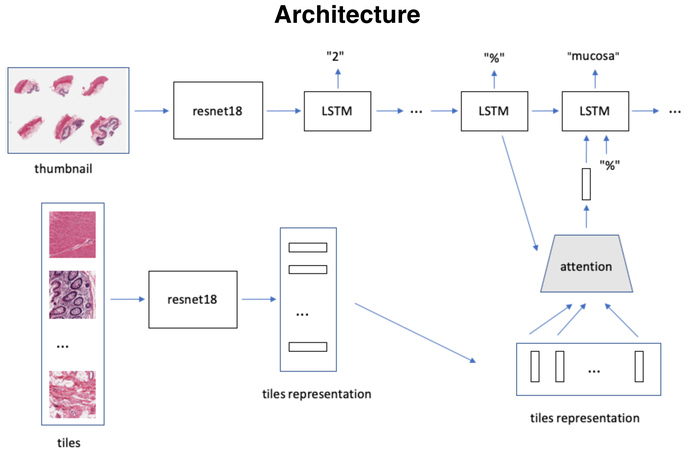


Figure 2: Overall architecture of PathCap. One ResNet-18 is used to extract visual features from the thumbnail of a histopathology image and pass it to the LSTM. The other ResNet-18 extracts features from randomly sampled tiles from different clusters of the histopathology image and passes them to the attention module and LSTM step by step.

Metric learning and clustering

A triplet t_j, t_k, t_l from a slide s. t_j is the anchor tile. t_k is a positive example. t_l is a negative example. The loss for training the tile autoencoder is

$$L(t_j, t_k, t_l) = \mu \cdot \max(d(t_j, t_k) - d(t_j, t_l) + m, 0) + d(t_j, D(e_j))$$

E is encoder and D is the decoder. $e_j = E(t_j)$. $d(\cdot,\cdot)$ represents the distance. m is the margin and μ is the factor for triplet loss.

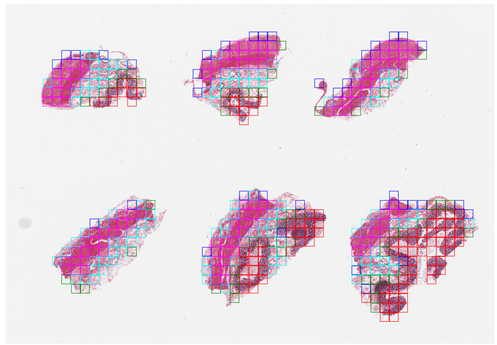


Figure 3: Example clustering visualization. Box color of each tile represents the cluster membership (K=5). The tile cluster colors demonstrate that tiles in a cluster are semantically coherent across and within pieces.

Figure 4: Example tiles used for triplet loss. (a) is the anchor tile showing colonic mucosa, (b) shows predominantly colonic mucosa, and (c) shows mostly smooth muscle (from muscularis propria). (b) and (c) correspond to positive and negative samples respectively for triplet loss.

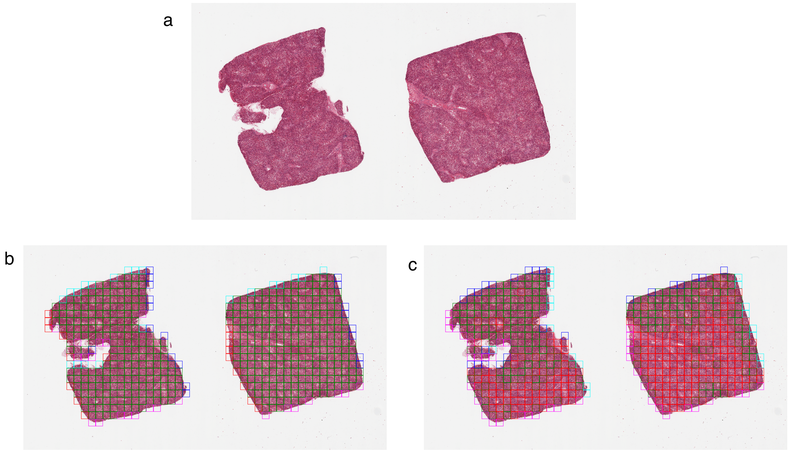


Figure 5: Example tile clustering (K=5) with triplet loss. (a) is the original slide. (b) and (c) show the tile clustering after we train the autoencoder with and without triplet loss respectively. Colors of the boxes show the cluster membership.

Performance of different models on GTEx test dataset

method	B-1	B-2	B-3	B-4	METEOR	ROUGE	CIDEr
Baseline	0.3822	0.2833	0.1996	0.1377	0.1958	0.4282	0.8936
PathCap	0.4046	0.2986	0.2114	0.1455	0.2059	0.4290	0.9038
Tiles-only	0.3944	0.2905	0.2040	0.1383	0.2032	0.4312	0.9003

Table 1: Baseline model only takes low-resolution thumbnails as input. For each step generating words, the model follows an attention mechanism and gives a weight for the spatial features extracted from thumbnails by ResNet-18. We also examined a version of PathCap that only used tiles and without access to a thumbnail view, and found that using tiles alone performed slightly better than the baseline model. Taken together, PathCap, which combines information from high-resolution tile and low-resolution thumbnail views performed the best.

Influence of triplet loss

Autoencoder loss	B-1	B-2	B-3	B-4	METEOR	ROUGE	CIDEr
Reconstruction only	0.3944	0.2878	0.2011	0.1381	0.2005	0.4219	0.8703
Reconstruction & triplet loss	0.4046	0.2986	0.2114	0.1455	0.2059	0.4290	0.9038

Table 2: In order to demonstrate the superiority of triplet loss on tile embeddings, we trained two autoencoders. One autoencoder was trained only with reconstruction loss. The other autoencoder was trained with reconstruction loss and triplet loss. The encoder part of the autoencoder was composed of two convolutional layers and two maxpooling layers. The output of the encoder (embedding) is of length 460. The decoder part contained three convolutional layers. The μ was set to 0.1, and the margin 0.001. We trained two separate PathCap models with the clusters using the representations from each of the two different autoencoders. Overall, we demonstrate both a qualitative improvement in tile-level clustering, and quantitative improvement in caption generation using metric learning.

Examples from test dataset

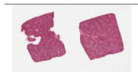
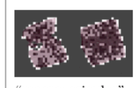
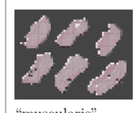
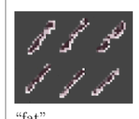
Slide	PathCap Prediction	Reference	Example
 Liver[a] a. GTEx sample ID: 13FLV-0326	2 pieces , diffuse macrovesicular steatosis involves 70 % of parenchyma	2 pieces ; includes portion of capsule (target is 1 cm below capsule) , mild steatosis , passive congestion , focal portal chronic inflammation	 "macrovesicular"
 Esophagus[b] b. GTEx sample ID: 13FTW-1926	6 pieces , up to <unk> ; all muscularis , good specimens	6 pieces ; well trimmed	 "muscularis"
 Skin[c] c. GTEx sample ID: 13NYS-0126	pieces ; well trimmed ; 5 % dermal fat	6 pieces ; <unk> epidermis (<unk>) , solar elastosis ; well trimmed , 10 % dermal fat	 "fat"
 Colon[d] d. GTEx sample ID: 13O3P-2326	6 pieces , mucosa up to 1mm , <unk> % thickness	6 pieces ; mucosa autolyzed ; muscularis preserved	 "mucosa"

Figure 5: Visualization of the PathCap method on four test slides from four different tissues. The last column shows some examples of attention weights when the model generates the corresponding tokens. White/bright indicates more attention weight, black/dark indicates less attention weight.}

Acknowledgements

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research, and the Center for Translational Data Science at the University of Chicago.