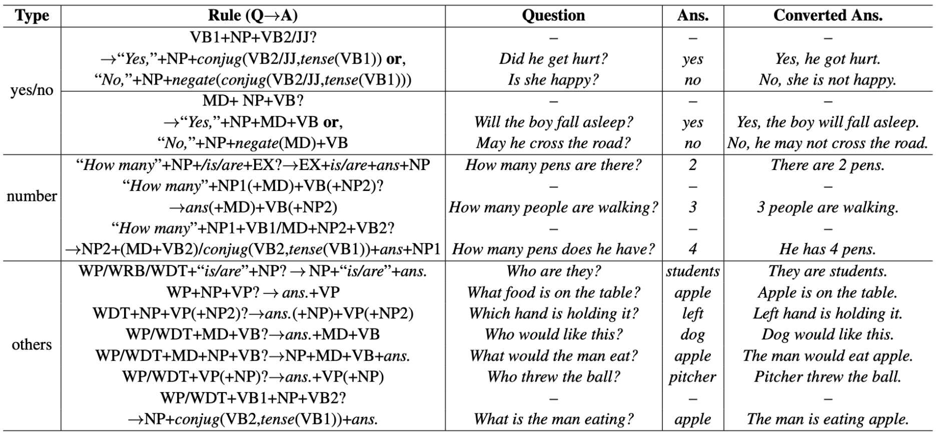
Generate Answer to Visual Questions in the Medical Domain: Phase 2

Niki Nezakati - 98522094
GitHub Repository

1 Creating the Full Sentence Dataset

In this research, I conducted a study that involved the utilization of the SLAKE [1] dataset, as discussed in phase one. The objective was to generate comprehensive answers from both single and multi-word answers present in the dataset. To accomplish this, I employed the proposed method outlined in the FSVQA [2] article. In order to convert the answers into complete sentences, I leveraged the NLTK language tool and pattern.en part-of-speech tagger. Taking into consideration the relevant question and answer, we applied the language rules introduced in the FSVQA article. A detailed description of these language rules can be found in the table 1 from this article. By implementing these rules on the SLAKE dataset, we successfully obtained a new dataset comprising answers in the form of complete sentences.



Type	Rule (Q→A)	Question	Ans.	Converted Ans.
yes/no	VB1+NP+VB2/JJ?	–	–	–
	→"*Yes,*"+NP+*conjug*(VB2/JJ,*tense*(VB1)) **or**,	*Did he get hurt?*	*yes*	*Yes, he got hurt.*
	"*No,*"+NP+*negate*(*conjug*(VB2/JJ,*tense*(VB1)))	*Is she happy?*	*no*	*No, she is not happy.*
	MD+ NP+VB?	–	–	–
	→"*Yes,*"+NP+MD+VB **or**,	*Will the boy fall asleep?*	*yes*	*Yes, the boy will fall asleep.*
	"*No,*"+NP+*negate*(MD)+VB	*May he cross the road?*	*no*	*No, he may not cross the road.*
number	"*How many*"+NP+*/is/are*+EX?→EX+*is/are*+*ans*+NP	*How many pens are there?*	*2*	*There are 2 pens.*
	"*How many*"+NP1(+MD)+VB(+NP2)?	–	–	–
	→*ans*(+MD)+VB(+NP2)	*How many people are walking?*	*3*	*3 people are walking.*
	"*How many*"+NP1+VB1/MD+NP2+VB2?	–	–	–
	→NP2+(MD+VB2)/*conjug*(VB2,*tense*(VB1))+*ans*+NP1	*How many pens does he have?*	*4*	*He has 4 pens.*
others	WP/WRB/WDT+"*is/are*"+NP? → NP+"*is/are*"+*ans.*	*Who are they?*	*students*	*They are students.*
	WP+NP+VP? → *ans.*+VP	*What food is on the table?*	*apple*	*Apple is on the table.*
	WDT+NP+VP(+NP2)?→*ans.*(+NP)+VP(+NP2)	*Which hand is holding it?*	*left*	*Left hand is holding it.*
	WP/WDT+MD+VB?→*ans.*+MD+VB	*Who would like this?*	*dog*	*Dog would like this.*
	WP/WDT+MD+NP+VB?→NP+MD+VB+*ans.*	*What would the man eat?*	*apple*	*The man would eat apple.*
	WP/WDT+VP(+NP)?→*ans.*+VP(+NP)	*Who threw the ball?*	*pitcher*	*Pitcher threw the ball.*
	WP/WDT+VB1+NP+VB2?	–	–	–
	→NP+*conjug*(VB2,*tense*(VB1))+*ans.*	*What is the man eating?*	*apple*	*The man is eating apple.*

Table 1: General conversion rules for converting captions to questions.

To do this, you should run the bash script located in the run folder named generate_fs_answers.sh. You can add one argument to this bash which is the output directory and you can specify that or leave it as the default value (data/full_sentence_data).

```
        bash run/generate_fs_answers.sh
```

After executing the provided bash script, you will have access to the resulting dataset. This dataset contains medical images along with corresponding question and full-sentence answer.

2 Word2Vec

In order to train Word2Vec on different data categories including Color, KG, Modality, Organ, Plane, Position, Shape, and Size, you should run the bash script located in the run folder named train_word2vec.sh. You should add the category names and number of iterations to this bash.

```
        bash run/dataset_tokenization.sh
```

3 Tokenization

Accurately processing questions entails breaking them down into smaller parts or tokens to enable deep neural networks to carry out the necessary computations. In the context of our experiments, where both the LXMERT and VisualBERT networks are utilized, we rely on BERT network tokenization to extract the features of the input text. This choice is made because both LXMERT and VisualBERT models are built upon the BERT network.

To do this, you should run the bash script located in the run folder named dataset_tokenization.sh. You should add the annotations and questions paths to this bash, in addition to the tokenizer set to [lxmert]. The output result would be saved in (data/train/tokenized), (data/val/tokenized), and (data/test/tokenized).

```
        bash run/dataset_tokenization.sh
```

4 Image Feature Extraction

In this study, we employed 4096-dimensional features extracted from the second fully connected layer (fc7) of the VGG model, which consists of 19 layers and was pre-trained on the ImageNet dataset, as our image features. The images were inputted into this model, with each image being treated as having 36 bounding boxes. Subsequently, each bounding box was mapped to a 1024-dimensional feature vector.

To do this, first we should resize the images. For this run the bash script located in the run folder named resize_images.sh. Afterwards, you should run the bash script located in the run folder named extract_image_features.sh. You can add one argument to this bash which is the output directory and you can specify that or leave it as the default value (data/imgs).

```
        bash run/resize_images.sh
        bash run/extract_image_features.sh
```

5 Language Model and Model Architecture

In this research, inspired by the work of "Generate Answer to Visual Questions with Pre-trained Vision-and-Language Embeddings" [3] we examine 4 model architectures. We utilise VisualBERT and LXMERT embedding space with two different generative decoder heads, including Attention RNNs and Transformers. In order to determine the most effective architecture for visual question answering in the medical domain, we conduct a comprehensive evaluation that combines qualitative and quantitative analysis along with a Human Error Analysis. We use BLEU, METEOR, and RougeL as our n-gram-based metrics, in addition to Average score and BERTScore as our embedding-based metrics.

To do this, you should run the bash script located in the run folder named train_genVQA.sh. You should add the encoder type, decoder type, rnn type, number of rnn layers, the bidirectionallity of the rnn, attention type, attention method, and number of heads and layers if you are utilizing the transformer decoder. The output result containg the scores would be saved in logs directory, named as the time and date of run.

```
        bash run/train_genVQA.sh
```

The table below shows the performance of the different models.

As we can see by the results, The utilization of the LSTM recurrent neural network in combination with the attention mechanism as a decoder exhibits superior performance compared to Transformers. One contributing factor to this observation is the limited size of the dataset. Transformers often struggle when confronted with small datasets due to their extensive parameterization and

Method		Word-based			Embedding-based	
Encoder	Encoder	BLEU	METEOR	ROUGE-L	Average Score	BERT Score
2*VisualBERT	1-LSTM+Bahdanau attention	**61.20**	**81.81**	**79.06**	**96.33**	**89.05**
	3-Transformer	12.30	14.11	15.00	61.89	33.89
2*LXMERT	1-LSTM+Bahdanau attention	56.78	77.25	75.37	95.10	85.36
	3-Transformer Decoder	24.94	52.26	53.89	70.59	69.10

Table 2: Results of our proposed models on the created dataset. The numbers in the decoder names mean the number of layers

high model capacity. These models, known for their data-hungry nature, require substantial training data to effectively capture complex patterns and generalize to unseen examples. In scenarios with small datasets and a constrained number of training samples, Transformers may lack the necessary diversity and coverage of underlying patterns, consequently leading to overfitting. Furthermore, the large number of parameters in Transformers exacerbates this issue by allowing the models to memorize the limited training samples, hampering their ability to generalize to new data.

Additionally, the quadratic computational complexity introduced by the self-attention mechanism in Transformers becomes particularly challenging to manage when working with smaller datasets. Consequently, the combination of data scarcity and computational inefficiency hampers Transformers' ability to learn meaningful representations from limited data. On the other hand, LSTM, along with attention, excels in capturing sequential dependencies and exhibits more effective handling of inputs with variable length. The recurrent nature of LSTM enables it to retain context from previous time steps, facilitating the modeling of temporal dependencies within the data. Furthermore, the attention mechanism selectively focuses on relevant parts of the input sequence, allowing the model to concentrate on crucial information. In contrast, Transformers operate in parallel and lack frequent connections, making them less suitable for tasks involving sequential data.

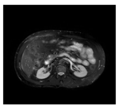
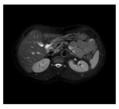
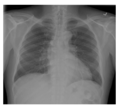
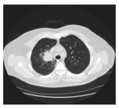
The improved performance of LSTM in conjunction with careful attention can be attributed to its robust architecture for capturing temporal patterns, which translates into enhanced performance in various tasks such as translation, text generation, and, in the context of this research, generating answers for image-related questions.

6 Human Analysis

In addition to the standard measurement criteria, this research incorporates a human evaluation process to assess the performance of the visual question answering system. For this purpose, a subset of elements was randomly selected from the dataset and subjected to evaluation using the superior model that was trained with the described architecture. Human evaluators carefully examined the answers generated by the system and categorized them into four distinct categories based on the type of mistakes observed:

- **EM** (Exact Match)
- **WA** (Wrong Answer): The model only generates incorrect VQA single/multi-word answers while the context and description are correct.
- **GE** (Grammatical Error)
- **WD** (Wrong Description): The model generates the correct VQA single/multi-word answer, but the description is wrong.

The table below illustrates some of these errors.

Type Error	Answer Generated	Answe and Question	Image
EM	VisualBERT-BahdanauLSTM: no, the picture does not contain spleen.	Q: does the picture contain spleen? A: no, the picture does not contain spleen.	
WA	VisualBERT-BahdanauLSTM: this image belongs to chest.	Q: which part of the body does this image belong to? A: this image belongs to abdomen.	
GE	VisualBERT-BahdanauLSTM: the abnormalitiesity located in the left lung.	Q: where is / are the abnormality located? A: the abnormalities are located in the left lung.	
WD	VisualBERT-BahdanauLSTM: lung is the largest organ in the picture.	Q: what is the main organ in the image? A: lung is the main organ in the image.	

References

[1] SLAKE: A Semantically-Labeled Knowledge-Enhanced Dataset for Medical Visual Question Answering. Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, Xiao-Ming Wu (2021). https://arxiv.org/abs/2102.09542

[2] The Color of the Cat is Gray: 1 Million Full-Sentences Visual Question Answering (FSVQA). Andrew Shin, Yoshitaka Ushiku, Tatsuya Harada (2016). https://doi.org/10.48550/arXiv.1609.06657

[3] Generate Answer to Visual Questions with Pre-trained Vision-and-Language Embeddings. Hadi Sheikhi, Maryam Hashemi, Sauleh Eetemad (2022). https://www.winlp.org/wpcontent/uploads/2022/11/73_Paper.pdf