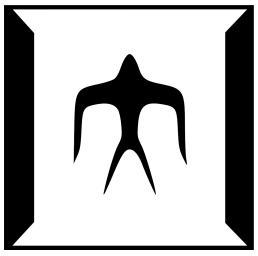
Image Captioning model for Low Resource Language Case study for Mongolian Language

Erdenebat Battseren
ID: 20B60029

Independent Research Project



Supervisor: Prof. Kayoko Nohara
Transdisciplinary Science and Engineering
Tokyo Institute of Technology
February, 2024

Contents

1 Introduction **5**
- 1.1 Background . . . 5
  - 1.1.1 Overview of Image Captioning . . . 5
  - 1.1.2 Image Understanding . . . 5
  - 1.1.3 Convolutional Neural Networks (CNNs) . . . 6
  - 1.1.4 CNN-base Pre-trained Models . . . 7
  - 1.1.5 Natural Language Understanding . . . 8
  - 1.1.6 Attention . . . 9
- 1.2 Low-resource Language . . . 9
  - 1.2.1 Existing Quantitative and Qualitative Evaluation . . . 9
  - 1.2.2 Mongolian Language . . . 10
  - 1.2.3 The Characteristics of Mongolian Language . . . 10
- 1.3 Research Objective . . . 11
- 1.4 Research Question . . . 11
- 1.5 Thesis Organization . . . 12

2 Related Works **12**
- 2.1 Existing Image Captioning Methods . . . 12
  - 2.1.1 Encoder-Decoder Framework . . . 12
  - 2.1.2 Attention Based Mechanism . . . 13
  - 2.1.3 Transformer Based Architectures . . . 14
- 2.2 Approaches in Low Resource Languages . . . 16
  - 2.2.1 Translation-based approaches . . . 16
  - 2.2.2 Alignment-based approaches . . . 17
- 2.3 Datasets and Evaluation Metrics . . . 17
  - 2.3.1 Datasets in Low Resource Languages . . . 17
  - 2.3.2 Evaluation Metrics . . . 18
- 2.4 Current State-of-the-art Benchmarks . . . 21

3 Image Captioning in Mongolian Language **21**
- 3.1 Significance of Building Baseline Model . . . 21
- 3.2 Overview of the Implemented Model . . . 23
- 3.3 Model Architecture . . . 23
  - 3.3.1 Encoder: Image Features . . . 23
  - 3.3.2 Decoder . . . 25
  - 3.3.3 Attention Mechanism . . . 25
  - 3.3.4 Beam Search . . . 26
- 3.4 Dataset Preparation . . . 27
  - 3.4.1 Development of dataset creating tool . . . 28
  - 3.4.2 Dataset Creation Process . . . 28
- 3.5 Dataset Preprocessing . . . 30
  - 3.5.1 Word Frequency Count and Word Map Creation . . . 30
  - 3.5.2 Image and Caption Processing . . . 30

3.5.3 Encoding and Saving Data . . . 30
3.6 Training . . . 30
3.7 Loss Function . . . 31
3.8 Computational Resource . . . 31

4 Results **32**
4.1 Quantitative Results . . . 32
4.1.1 Discussion on the Quantitative Result . . . 33
4.2 Qualitative Results . . . 34
4.2.1 Conclusion . . . 42

5 Discussion and Conclusion **43**
5.1 Thesis Contributions . . . 43
5.2 Future work . . . 43
5.3 Conclusion . . . 43

List of Figures

1 Convolutional operation with kernel (Source: Link) . . . 6
2 Padding layer (Source: Link) . . . 6
3 The architecture of CNN. (Source: Link) . . . 7
4 Altaic-region Language Family . . . 10
5 Comparison between traditional approach and modern approaches . . . 14
6 Scaled Dot-Product Attention and Multi-Head Attention (From the paper [17]) 15
7 Transformer Architecture Originally proposed in 2017 [17] . . . 15
8 Summary of existing multilingual (Source: Hideki Nakayama et al., [38]) . . . 17
9 METEOR Algorithm: Alignment (Source: Link) . . . 19
10 Current State-of-the-art Benchmarks . . . 21
11 Image Encoder . . . 23
12 RNN-based Decoder . . . 25
13 Attention Mechanism . . . 26
14 Combined architecture . . . 26
15 Beam Search Algorithm Example: When $B = 3$ . . . 27
16 Annotation tool developed for this Research . . . 28
17 Flickr30K_MN Dataset creation workflow . . . 29
18 Quantitative Results: Comparison between baseline model, attention-based model, and machine translation . . . 32
19 Qualitative Evaluation: Example 1 (Beam size = 3) . . . 35
20 Qualitative Evaluation: Example 2 (Beam size = 3) . . . 35
21 Qualitative Evaluation: Example 3 (Beam size = 3) . . . 36
22 Qualitative Evaluation: Example 4 (Beam size = 3) . . . 37
23 Qualitative Evaluation: Example 6 (Beam size = 3) . . . 37
24 Qualitative Evaluation: Example 7 (Beam size = 3) . . . 38
25 Insufficient Example: Mirror Reflection . . . 39

26 Poor Example: Not Attending relevant part of the image 39
27 Poor Example: Not recognizing objects 40
28 Poor Example 4: Poor alignment in image and text 40
29 Cultural specific: Mongolian wrestling 41
30 Cultural specific: Mongolian Greeting 41
31 Cultural specific: Mongolian Game 42

Listings

1 Encoder Implementation 23
2 Hyperparameters for the training 30

List of Tables

1 Common CNN-Based Pretrained Models with Details 8
2 Interpretation of BLEU Score 19

Abstract

The field of image understanding has experienced significant advancements in recent years, primarily due to the emergence of innovative architectural approaches. However, it's important to note that the majority of state-of-the-art achievements are predominantly seen in English, due to the abundant data availability in the language. In contrast, the scarcity of data in other languages presents considerable challenges for effectively extending these advancements to multilingual contexts. Addressing this issue, the focus of this research is to enhance model performance in languages where data availability is limited. A key part of the study involved developing a Mongolian version of the Flickr30K dataset, achieving the initial benchmark results, and proposing a new model architecture. By employing strategies like transfer-learning and the attention mechanism, this research not only established the first benchmark results for Mongolian in both qualitative and quantitative analyses but also laid the groundwork for future enhancements in this area.

1 Introduction

1.1 Background

1.1.1 Overview of Image Captioning

Image captioning is an interdisciplinary field that combines techniques from computer vision and natural language processing (NLP). It involves creating descriptive captions for images, requiring the system to understand the content of an image and express it in natural language. One notable example with high social applicability is designing assistants for the visually impaired. This task is considerably harder than well-studied image classification or object recognition tasks because the model not only needs to recognize the objects inside an image, but also needs to express how these objects relate to each other as well as their attributes, and activities involved [1]. However, over the years, this field has seen considerable advances with the advent of deep learning techniques. Initially, image captioning relied on traditional machine learning methods, utilizing feature extraction techniques such as Scale-Invariant Feature Transform (SIFT) [2], Local Binary Patterns (LBP) [3], and the Histogram of Oriented Gradients (HOG) [4]. However, due to the computational limitations and complexity of the features in large datasets, the traditional approaches are replaced by deep learning techniques.

1.1.2 Image Understanding

The backbone of the image captioning task is image understanding, which builds upon the outcomes of image processing and image analysis, aims to interpret the significance of an image at a higher level. This interpretation seeks to provide semantic information that aligns closely with human cognition, assisting in decision-making and guiding actions based on scene comprehension. When integrated with Natural Language Processing (NLP), this process evolves into Image Captioning. Image Captioning involves converting visual components into expressions in natural language. Thus, visual understanding requires not only object recognition but also the thorough extraction of details pertaining to each object and their interrelationships.

Below shows the historical perspectives about the advancement in visual understanding:

- **Early Stages (Until 2000s):** Initially, image understanding relied on rule-based systems and simple pattern recognition techniques such as Sobel operator or Canny edge detector.
- **Introduction of Machine Learning (2000s - 2010):** With the rise of machine learning, techniques such as Support Vector Machine (SVM) or decision trees began to be used for tasks such as object recognition and classification.
- **Deep Learning Revolution (2010s and onward):** The introduction of deep learning, especially Convolutional Neural Networks (CNN), marked a significant shift. CNNs are deep neural network architectures designed for working with images. The success of models like AlexNet led to the development of more advanced architectures like VGG, ResNet, and Inception, which are also used in this research. These models are trained on huge datasets such as ImageNet and could be fine-tuned for specific tasks. The excellent performance of these models lowered the barrier to vision-based tasks.

- **Object Detection and Beyond (2020s onwards)** Advances in object detection frameworks like Vision Transformer, abd YOLOv8 have expanded the scope of image understanding. These models can localize and classify multiple objects within an image, a step beyond simple classification.

1.1.3 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are the encoder part of the image captioning model. CNNs are able to automatically learn the features from raw data [5]. Simply speaking, CNN layers are feature extractors where the input layers extract low-level features from raw data, and the fully connected layers use those features to predict target value or class label. This architecture makes them extremely effective for tasks such as image and video recognition, image classification. CNNs are composed of several convolutional layers and sub-sampling layers that are followed by one more fully connected layers at the end. The layers include:

- **Convolutional layers:** These layers apply a set of learnable filters (kernel) to the input. Each filter activates certain features from the image, such as edges or textures. There is additional parameter called stride, which controls the movement of the filter. For example, if the stride value is 1, the filter will take one-by-one. If the stride value is 2, the kernel will take values skipping every two pixels.

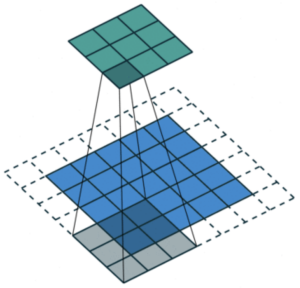


Figure 1: Convolutional operation with kernel (Source: Link)

- **Padding:** refers to the number of pixels added to an image when it is being processed by kernel.

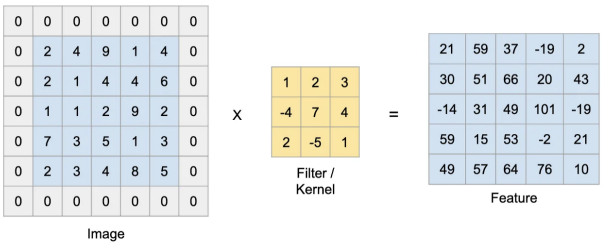


Figure 2: Padding layer (Source: Link)

- **Pooling layers:** is a technique for generalizing features extracted by convolutional filters and helping the network recognize features independent of their location in the image.
- **Fully connected layer:** or dense layer consists of the weights and biases along with the neurons and is critical to make sense of the features extracted by the convolutional layers.
- **Activition function:** decides whether a neuron should be activated or not. Function can decide whether the input is important or not in the process of prediction. The common activation functions are ReLU, Softmax, tanH, and the Sigmoid functions.

Below is the integrated architecture of the CNN model.

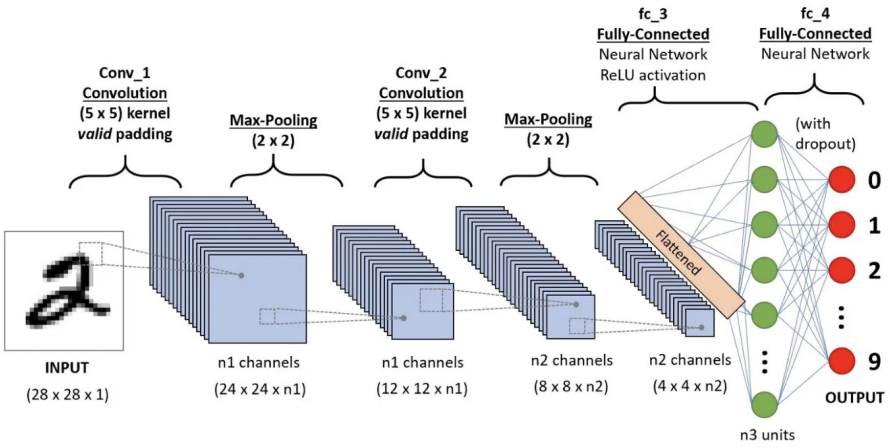


Figure 3: The architecture of CNN. (Source: Link)

1.1.4 CNN-base Pre-trained Models

Nowadays, it is rare to train CNN models from scratch in general tasks, but rather utilize pretrained models due to their versatility and excellent performance. Table [1] shows some of the most commonly used CNN-based pretrained models, where some of them used in this research.

Model	Details	Reference
AlexNet	One of the first deep CNNs; breakthrough performance in ImageNet challenge.	Krizhevsky et al., 2012 [7]
VGGNet	Known for depth (VGG16, VGG19); simple and effective architecture.	Simonyan and Zisserman, 2014 [8]
GoogLeNet (Inception v1)	Introduced inception module; efficient with lower computational demand.	Szegedy et al., 2015 [9]
ResNet	Introduced residual connections; variants include ResNet-50, ResNet-101, and ResNet-152.	He et al., 2016 [10]
Inception-v3	Improvement over Inception with various optimizations.	Szegedy et al., 2016 [11]
DenseNet	Features densely connected layers for improved information flow.	Huang et al., 2017 [12]
MobileNet	Optimized for mobile devices; balances efficiency and accuracy.	Howard et al., 2017 [13]
EfficientNet	Uniform scaling of network dimensions; efficient and scalable.	Tan and Le, 2019 [14]

Table 1: Common CNN-Based Pretrained Models with Details

1.1.5 Natural Language Understanding

Another crucial component of the image captioning task is Natural Language Understanding (NLU), which refers to the subset of Natural Language Processing (NLP) that deals with machine reading comprehension. It features the machine's ability to understand and interpret human language. NLU is concerned with nuanced aspects of language such as context, intent, emotions. Recurrent Neural Network (RNN) [15] and Long-Short-Term-Memory (LSTM) [16] were the dominating architectures before the current paradigm in Transformer architecture [17].

In traditional encoder-decoder frameworks, a CNN encoder extracts image features that are then fed into an RNN-based decoder, such as an LSTM or GRU. More recently, a transformer-based decoder is employed for this purpose. RNN-based model would generate a word, updates its hidden state, and uses this updated state to predict the next word. A transformer-based model on the other hand would use the CNN output as part of its input sequence for the encoder or directly to the decoder. The model applies self-attention mechanisms to weigh the importance of each part of the input throughout the sequence.

1.1.6 Attention

The attention mechanism, a pivotal component in tasks such as machine translation and image captioning, was introduced to improve upon the limitations of earlier sequence-to-sequence models. The early models processed input sequences (such as sentences) and converted them into a fixed-length context vector, a condensed representation. Then, they generated the output sequence from this vector. However, these models faced challenges with longer input sequences because the capacity of the fixed-length context vector was limited, making it difficult to adequately represent all the information from longer sequences.

Bahdanau et al. [18] proposed a solution to this in their seminal paper "Neural Machine Translation by Jointly Learning to Align and Translate," which introduced the concept of attention that allows the model to automatically learn to focus on different parts of the input sequence when predicting a certain part of the output sequence, thus creating a dynamic context vector instead of a fixed one. The attention mechanism can be categorized mainly into two types: soft and hard attention. Soft attention computes a weighted sum of all input states, with the weights designed to sum to one, through activation functions such as softmax. This method is differentiable and allows fort efficient training using backpropagation techniques. Hard attention, on the other hand, selects a subset of input states to focus on and is often non-differentiable, requiring different training strategies like reinforcement learning. In this research, soft attention mechanism is utilized due to the above stated reasons.

1.2 Low-resource Language

In the context of image understanding, a "low-resource language" refers to languages for which there is a limited amount of available data, particularly annotated data, necessary for training deep learning models like those used in image captioning tasks. This limitation presents a significant challenge because most state-of-the-art image captioning models are data-intensive, relying on large datasets to achieve high performance.

The research paper "Improving Captioning for Low-Resource Languages by Cycle Consistenc" [19] addresses this challenge by proposing a method to leverage high-resource languages, like English, to enhance captioning performance in low-resource languages. The approach involves using a pre-trained English caption model to generate English captions, which are then used along with the image to generate captions in the low-resource language. This method is designed to benefit from large monolingual English caption datasets and has shown to outperform existing methods in terms of common evaluation metrics.

1.2.1 Existing Quantitative and Qualitative Evaluation

Quantitative Analysis

Quantitative analysis typically involves the use of performance metrics to objectively evaluate the accuracy and relevance of the generated captions. Common metrics include BLEU [21], METEOR [23], ROUGE [24], and CIDEr [22]. These metrics compare the machine-generated captions against a set of human written reference captions to quantify how closely the generated text matches the reference in terms of structure, content, and meaning.

Qualitative Analysis

Qualitative analysis focuses on the aspects that are not easily quantifiable, such as the naturalness, coherence, relevance and overall quality of the generated captions. It often involves human evaluators assessing the captions or detailed examination of the model's behavior in various scenarios. Qualitative analysis is crucial for understanding the model's capabilities in generating contextually and semantically appropriate captions, dealing with complex scenes, and handling nuances of language. In this research, attention visualization is used as an qualitative analysis to see if the model is focusing on the specific part of the picture.

1.2.2 Mongolian Language

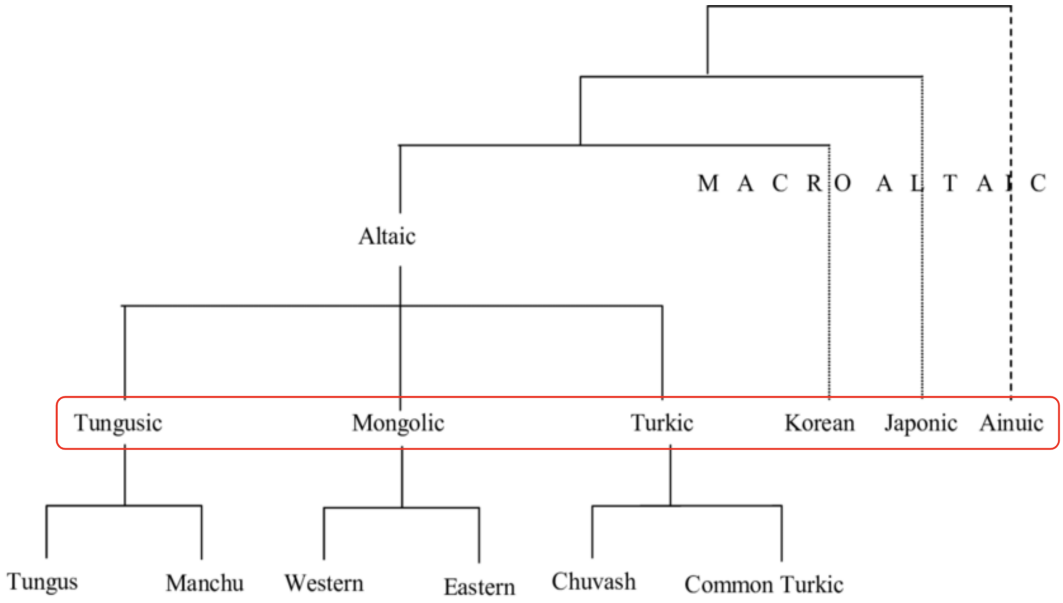


Figure 4: Altaic-region Language Family

Mongolian belongs to the Mongolic language family. The Mongolic languages are a group of languages spoken in East-Central Asia, primarily in Mongolia and surrounding areas. Mongolian is often classified as part of the Altaic language family (Figure [4]), a grouping that has been debated among linguists. The Altaic hypothesis proposes a language family consisting of Turkic, Mongolic, Tungustic, Koreanic, Japonic, and Ainuic languages, based on observed similarities in vocabulary, morphology, syntax, and phonology. Some linguists argue for a genetic relationship between these language, tracing their origins to the steppes near the Altai Mountains [25].

1.2.3 The Characteristics of Mongolian Language

Some of the unique characteristics of Mongolian languages are:

- **Agglutinative Structure:** Mongolian is agglutinate, meaning it forms words by combining a series of affixes with root word. This structure allow for the expression of complex ideas in single, compound words.

- **Vowel Harmony:** Mongolian features vowel harmony, a trait common in Turkic and Altaic languages. This means the vowels within a word must be harmonically consistent (wither all front or all black vowels), affecting suffixation.
- **Subject-Oject-Verb (SOV) Order:** The typical word order in Mongolian is Subject-Object-Verb, unlike English's Subject-Verb-Object. This affects how sentences are structured and understood.
- **Cyrillic Script:** Since the 1940s, Mongolian has primiarly been written in Cyrillic alphabet, with the traditional Mongolian script still used in Inner Mongolian and China.
- **Phonology:** Mongolian's phonetic system includes sounds not found in English, which can pose pronunciation challenges for learners

These linguistic features, particularly the agglutinative structure and vowel harmony, give the Mongolian language its distinct character and complexity. Understanding these characteristics is crucial for effectively developing language processing tools, such as those used in image captioning models, especially when adapting these tools from languages with very different structures, like English. Chapter 4, Results, shows some of the difference in linguistic features on the qualitative analysis.

1.3 Research Objective

The main objective of this thesis is to understand the characteristics of low-resource image captioning models through the development of the first image captioning model tailored for the Mongolian language. The establishment of a baseline model is expected to serve as a reference point, enabling further research to more effectively determine promising approaches for enhancing performance in the low-resource languages. The findings of this research are expected to provide insights for development proposals related to languages linguistically similar to Mongolian, such as those in the Altaic language family.

Additionally, there are a small number of researchers in this field, and this research is expected to foster community engagement by encouraging other researchers to contribute to the topic specially those who are trying to generalize these tasks in multilingual context.

1.4 Research Question

Based on the research objective, following research questions are raised:

- What are the specific challenges in developing an image captioning model for the low-resource language?
- What linguistic aspects of the Mongolian language are most challenging to capture in an image captioning model?
- How can insights from developing this model inform linguistic research, especially regarding Mongolian and similar languages?

- What are the potential areas for further research in improving image captioning models for low-resource languages based on the findings from the Mongolian model?

These questions will be discussed in Chapter 5, based on the findings of the research.

1.5 Thesis Organization

This thesis is organized into five main chapters, each focusing on different aspects of developing and evaluating an image captioning model for the Mongolian language. Chapter 1, Introduction, sets the stage for the thesis by providing necessary background information. It starts with an overview of image captioning, covering key concepts like image understanding, CNNs, natural language understanding, and attention mechanisms. Chapter 2, reviews existing literature and methods in the field of image captioning. Chapter 2 also examines existing datasets and evaluation metrics, and the current state-of-the-art benchmarks in image captioning. Chapter 3 discusses the actual development of the image captioning model for the Mongolian language. It begins by explaining the significance of building a baseline model and provides an overview of the implemented model. Chapter 4, presents the outcomes of the research, both quantitative and qualitative. It includes a discussion on the results, providing insights into how the model performed and any significant findings that emerged from the analysis. Chapter 5, the final chapter concludes the findings of the thesis, discussing its broader implications and contributions to the field.

2 Related Works

2.1 Existing Image Captioning Methods

In this section, various methodologies that have been developed over time for image captioning are discussed. For low-resource languages, it's important to understand the evolution of these methods to identify which ones can be adapted or improved upon languages with less data availability.

2.1.1 Encoder-Decoder Framework

In the deep learning based methods, due to the early success of language modeling, encoder-decoder framework was born. Bengio et. al, in the paper "Show and Tell" [1] used CNNs to encode an image into an embedding space such that it could be then used as an input to a downstream decoder to generate textual tokens.

Encoder:

- For the encoder, the typical architecture would consist of a pre-trained CNNs models such as ResNet or VGG.
- The role of the CNN is to extract features from the input image. These features are a representation of the image in a form that the model can process - typically a vector or a set of vectors.

- The CNN goes through layers of convolutions, pooling, and possibly other operations to gradually abstract the raw pixel data into higher-level features.

Decoder:

- Often a Recurrent Neural Network (RNN) or a model with an attention mechanism (like the Transformer model), which takes the output of the encoder and generates the text output.
- In image captioning, the decoder translates the feature representation provided by the encoder into a human-readable caption.
- The RNN or similar architecture in the decoder processes the features sequentially, often one word at a time, to build up the caption. It takes into account not only the features from the image but also the context from the previously generated words to predict the next word in the sequence.

2.1.2 Attention Based Mechanism

"Show, Attend, and Tell" by Vinyals et el., 2015 [28] integrates attention mechanism into the encoder-decoder framework. Attention mechanism was first developed in machine translation model in 2015 by Bengio et al., [29]. Due to the attention model's mechanism, which learns to focus on various aspects of the input image while generating captions, the adoption of attention mechanisms has become common across numerous captioning models. Lu et al., [30] for instance, implemented adaptive attention mechanism, which decides whether to focus on the image based on the current context and the generated word token. Another influential methodology was a novel bottom-up and top-down approach, leveraging object detectors to extract visual features, a shift from the traditional use of the final convolutional layers.

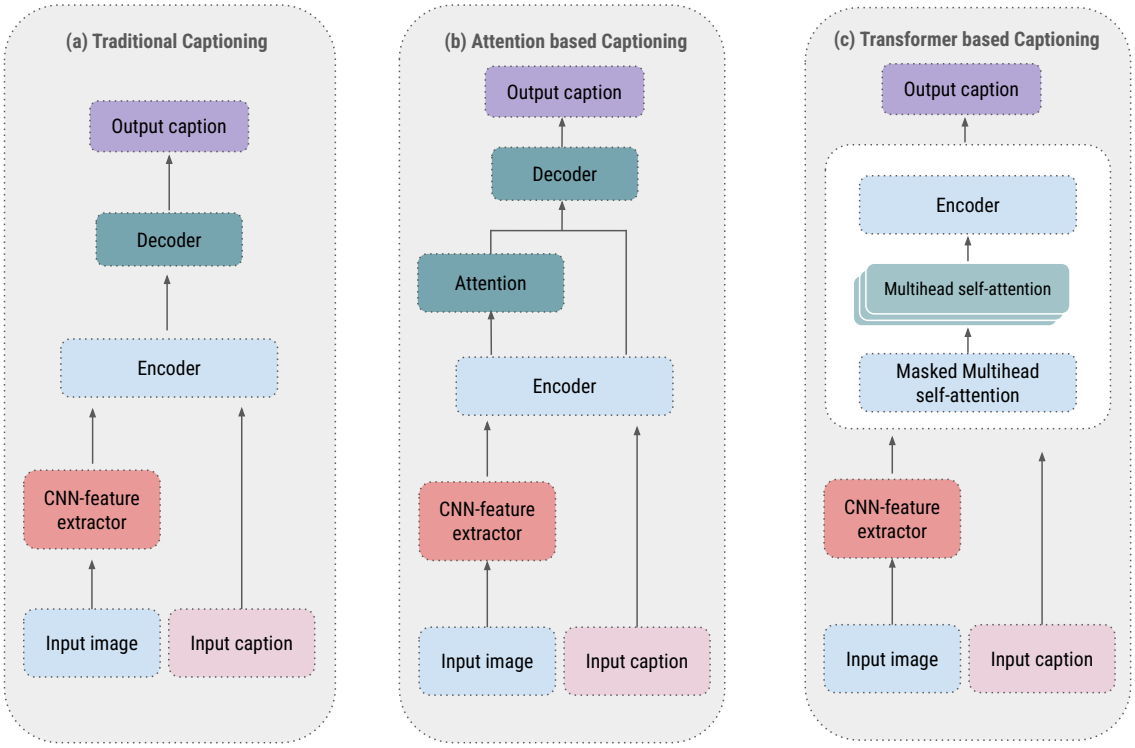


Figure 5: Comparison between traditional approach and modern approaches

2.1.3 Transformer Based Architectures

Originating from the seminal paper "*Attention is All You Need*" by Vaswani et al. [17], transformer models have revolutionized attention-based methodologies in machine learning. These models are now fundamental to the ongoing advancements in artificial intelligence, particularly in natural language processing. They underpin the framework of several cutting-edge AI models, including the popular GPTs [32], the newly released Gemini [33], the BERT [31], among others, showcasing their versatility and impact across various applications.

Scaled Dot-Product Attention: The inputs consists of queries and keys of dimension d_k, and values of dimension d_v. In practice, the queries, keys, and values are computed simultaneously, packed into a matrix Q, K, V:

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

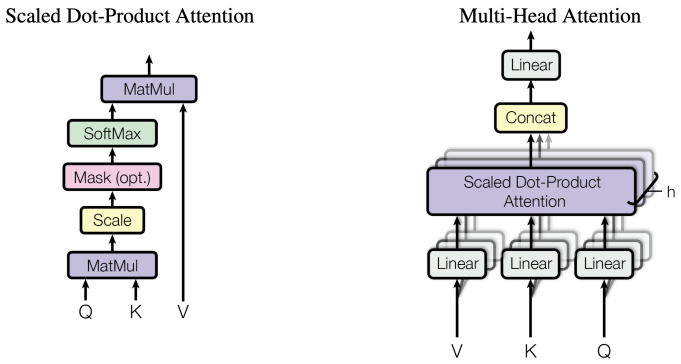


Figure 6: Scaled Dot-Product Attention and Multi-Head Attention (From the paper [17])

hello

Multi-Head Attention: This is slightly different kind of attention

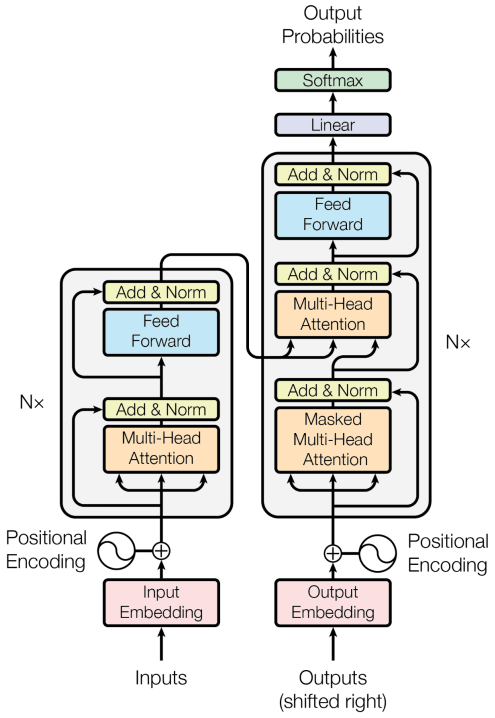


Figure 7: Transformer Architecture Originally proposed in 2017 [17]

BERT: Bidirectional Encoder Representations from Transformers

Devlin et al. [31] BERT is primarily known for its use in natural language processing, it has also been adapted for multimodal tasks that include both image and text, thus enhancing the performance of image captioning models. BERT's powerful language understanding capabilities can be leveraged in a two-step image captioning process. Its extensive pre-training on language data allows it to bring a deep understanding of language nuances, grammar, and syntax to these tasks. For low-resource languages, such as Mongolian, where there

is limited training data for language tasks, BERT can significantly enhance performance. Tuguldur, Erdene-Ochir, Gunchinish, Sharavsambuu, and Bataa, Enkhbold [34] released "*BERT Pretrained Models on Mongolian Datasets*", consisting of a vocabulary size of 30,000 words. Future work is expected to incorporate a pretrained Mongolian BERT model into the system.

GPT: Generative Pretrained Transformer

Models like GPT-2, GPT-3, and GPT-4 [32], developed by OpenAI, utilize transformer architecture with a focus on language generation. They can be fine-tuned for specific tasks such as image captioning, and visual question answering when paired with visual inputs. A comparison in traditional approach, RNN attention-based approach, and transformer-based approach can be seen from the Figure [5]. The decoder can take an input from the CNN embeddings or more recently from transformer-based visual encoders such as Vision Transformer. [35]

Vision Transformer

Vision Transformers (ViTs) have been increasingly used in image captioning tasks, demonstrating their versatility and effectiveness. In the context of image captioning, Vision Transformers are employed to process visual data, capturing complex patterns and features that are essential for generating accurate and relevant captions. A notable approach in this field is the CPTR model [36], as detailed in the paper "*CPTR: Full Transformer Network for Image Captioning*". This model is a pure Transformer-based architecture, designed specifically for image captioning. It uses the Vision Transformer to process the visual input and a language model for generating the captions. The CPTR model demonstrates significant improvements over traditional CNN-based models in captioning tasks, showcasing the effectiveness of Transformers in handling both visual and linguistic data.

2.2 Approaches in Low Resource Languages

While advancements in image understanding have predominantly been in English, researchers have begun addressing challenges such as limited data availability and the lack of pre-trained models for other low-resource languages with alternative approaches. A review of existing literature shows that (also noted by papers [38], [19]) these efforts typically fall into two main strategies: translation-based and alignment-based approaches. The translation-based approach focuses on converting non-English data into English for utilizing existing models and methods, while the alignment-based approach seeks to directly align multilingual data with images, bypassing the need for translation.

2.2.1 Translation-based approaches

The translation-based approach in image understanding involves translating non-English textual data into English to leverage existing models and methodologies developed primarily for the English language. This method relies on the assumption that translating the textual content will provide access to the wealth of English-language tools and resources, thereby enabling effective image understanding.

2.2.2 Alignment-based approaches

The alignment-based approach directly correlates multilingual data with images, bypassing the need for translation. This method aims to establish a direct relationship between the visual content of images and text in various languages. It seeks to understand and interpret images in conjunction with text in the original language, thus maintaining the cultural and linguistic context. This approach is especially beneficial for languages with limited resources, as it doesn't depend on the availability of high-quality translation systems.

2.3 Datasets and Evaluation Metrics

2.3.1 Datasets in Low Resource Languages

Hideki Nakayama, Akihiro Tamura, Takashi Ninomiya [38] in the paper "A Visually-Grounded Parallel Corpus with Phrase-to-Region Linking" summarized the datasets created in low-resource languages. It can be seen that many multilingual datasets are proposed by extending the original English datasets such as Flickr30k [26], MS-COCO [39]. It can be seen that most of the datasets do not include phrase-to-region linking and big datasets like MSCOCO are usually machine translated due to time and resource limitations.

Dataset	Languages	Source	# Imgs	# Sent.	Human Trans.	Phrase-Region Linking
IAPR-TC12 (Henning et al., 2006)	German, English	Web images	20,000	20,000	Yes	No
Pascal Sentences JP (Funaki and Nakayama, 2015)	Japanese, English	Pascal Sentences	1,000	5,000	Yes	No
Flickr8k-CN (Li et al., 2016)	Chinese, English	Flickr8k	8,000	40,000	Partial (test set)	No
Flickr30k-CN (Lan et al., 2017)	Chinese, English	Flickr30k	31,783	158,915	Partial (test set)	No
Multi30K (Translations) (Elliott et al., 2016)	German, French, Czech, English	Flickr30k	31,014	31,014	Yes	No
YJ Captions (Miyazaki and Shimizu, 2016)	Japanese, English	MS-COCO	26,500	131,740	No	No
STAIR Captions (Yoshikawa et al., 2017)	Japanese, English	MS-COCO	164,062	820,310	No	No
MIC test data (Rajendran et al., 2016)	German, French, English	MS-COCO	1,000	5,000	No	No
Bilingual caption (Hitschler et al., 2016)	German, English	MS-COCO	1,000	1,000	Yes	No
COCO-CN (Li et al., 2019)	Chinese, English	MS-COCO	20,342	27,218	Partial (test set)	No
Hindi Visual Genome (Parida et al., 2019)	Hindi, English	Visual Genome	31,525	31,525	Yes	Yes (one-to-one)
Flickr30k Entities JP	**Japanese, English**	**Flickr30k Entities**	**31,783**	**63,566**	**Yes**	**Yes**

Figure 8: Summary of existing multilingual (Source: Hideki Nakayama et al., [38])

2.3.2 Evaluation Metrics

BLEU: Bilingual Evaluation Study - n-gram precision
BLEU metric is a commonly used algorithm for evaluating the quality of machine-generated text, especially in the context of machine translation.
Mathematical Description Mathematically, the BLEU score is defined as:

$$\text{BLEU} = \underbrace{\min\left(1, \exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right)}_{\text{brevity penalty}} \underbrace{\left(\prod_{i=1}^{4} \text{precision}_i\right)^{1/4}}_{\text{n-gram overlap}} \tag{2}$$

with

$$\text{precision}_i = \frac{\sum_{\text{snt} \in \text{Cand-Corpus}} w_t^i}{\sum_{\text{snt'} \in \text{Cand-Corpus}} \sum_{i' \in \text{snt'}} m_{\text{cand}}^{i'}} \tag{3}$$

where:

- m_i^{cand}: is the count of i-gram candidate matching the reference translation
- m_{ref}^i : is the count of -gram in the reference translation
- w_i^t : is the total number of i-grms in candidate translation

There are some unique properties of BLEU score:

- **BLEU is a corpus based metric:** The BLEU metric performs poorly when used to evaluate individual sentences. For example, the score can be low even though they capture most of the meaning. Due to the fact that n-gram statistics for individual sentences are less meaningful, the score is accumulated over an entire corpus.
- **No distinction between content and function words:** BLEU does not distinguish between content word and function words. For instance, a missed function word such as "a" or "the" gets same penalty as if the name "Japan" were incorrectly replaced by "Mongolia."
- **Not suited for capturing the meaning and grammatical correctness:** The drop of word such as "do" or "don't" can change the polarity of the sentence. Also, taking only the n-grams into account with $n \leq 4$ ignores the long range dependencies and BLEU often imposes small penalties for grammatically not correct sentences.
- **BLEU is inconsistent with different languages:** Each language has its own characteristics, German for instance, has a free word order compared to English and Romance languages [40], and often not recommended for German.

The interpretation of BLEU scores is the following:

BLEU Score	Interpretation
< 10	Almost useless
10-19	Hard to get the gist
20-29	The gist is clear, but has significant grammatical errors
30-40	Understandable to good translations
40-50	High quality translations
50-60	Very high quality, adequate
> 60	Quality often better than human

Table 2: Interpretation of BLEU Score

However, a description generated from image captioning can be different from machine translation and the scores do not always follow the theoretical calculation. BLEU score above 40 is extremely rare and often better than human translators. In practice, a score above 30 is considered very good and above 20 to 30 is considered good. These characteristics will be discussed in the later part with alternative quantitative evaluations such as METEOR CIDEr.

METEOR: Metric for Evaluation of Translation with Explicit Ordering
METEOR is designed to address some of the limitations of BLEU by incorporating additional features and considering word order and synonyms.Through stemming and synonym matching, METEOR is designed to produce correlation with human judgements. This is the main difference between BLEU metric where BLEU seeks correlation at corpus level.

Algorithm Similar to BLEU metric, the basic unit of of evaluation is the sentence. The algorithm first creates an alignment reference sentence and candidate sentence. The alignment is essentially a set of mappings between unigrams. An example can be seen from the Figure [9].

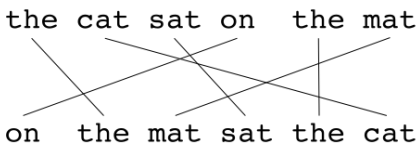


Figure 9: METEOR Algorithm: Alignment (Source: Link)

Once alignment is computed, the score is calculated as follows.

Precision (P): The number of matched unigrams in the candidate sentence divided by the total number of unigrams in the candidate sentence.

$$P = \frac{\text{number of matched unigrams}}{\text{total unigrams in candidate}} \tag{4}$$

Recall (R): The number of matched unigrams in the candidate sentence divided by the total number of unigrams in the reference sentence.

$$P = \frac{\text{number of matched unigrams}}{\text{total unigrams in reference}} \tag{5}$$

Harmonic Mean (Fmean): METEOR uses the harmonic mean of precision and recall weighted 9 times more than the precision.

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9 \cdot P} \tag{6}$$

Penalty (p): The penalty is based on the number of chucks (c) and the number of matched unigrams (m).

$$p = 0.5 \cdot \left(\frac{c}{m}\right)^3 \tag{7}$$

Final METEOR Score: The final METEOR is then calculated by Fmean penalized by the fragmentation penalty.

$$\text{METEOR} = (1 - p) \cdot F_{\text{mean}} \tag{8}$$

Generally, METEOR is considered more nuanced alternative to BLEU for evatluating the quality of generated captions. Unlike BLEU, METEOR considers both precision and recall, giving a balanced measure that accounts for the coverage of content between candidate and reference sentences.

CIDEr: Consensus-based Image Description Evaluation
Unlike BLEU and METEOR, which are general text metrics, CIDEr is specifically designed for image captioning tasks by Ramashkrishna et el., [22]. It focuses on capturing the quality and relevance of image descriptions. While METEOR focuses on semantic meaning with flexible matches, CIDEr goes a step further by focusing on consensus. It measures the similarity of a candidate caption to a set of reference captions that are written by humans.

Term Frequency Inverse Document Frequency (TF-IDF): The authors proposed TF-IDF [22] where it weights each n-gram in the reference and candidate captions. In the original paperm TF-IDF is defined as:

The number of times an n-gram w_k occurs in a reference sentence s_{ij} is denoted by $h_k(s_{ij})$ or $h_k(c_i)$ for the candidate sentence c_i. The TF-IDF is then calculated as:

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{w_l \in \Omega} h_l(s_{ij})} \log\left(\frac{|I|}{\sum_{I_P \in I} \min\left(1, \sum_q h_k(s_{pq})\right)}\right) \tag{9}$$

where Ω is the vocabulary of all n-grams and I is the set of all images. The first term measures the TF of each n-gram w_k, and the second term measures the rarity of w_k using IDF.

The CIDEr score n-grams of length n is computed using the average cosine similarity between the candidate sentence and the reference sentence, which accounts for both precision and recall.

$$\text{CIDEr}_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\left\|g^n(c_i)\right\| \left\|g^n(s_{ij})\right\|} \tag{10}$$

The scores from n-grams of varying lengths can be expressed as follows:

$$\text{CIDEr}(c_i, S_i) = \sum_{n=1}^{N} w_n \text{CIDEr}_n(c_i, S_i) \tag{11}$$

Three different evaluation metrics including BLEU, METEOR, and CIDEr will be used as a quantitative evaluation for this research.

2.4 Current State-of-the-art Benchmarks

Current state-of-the-art benchmarks trained on the COCO dataset.

Current State-of-the-art Benchmark Trained on COCO Dataset

Rank	Model	BLEU-4	CIDEr	METEOR	SPICE	ROUGE-L	Year	Paper
1	mPLUG	46.5	155.1	32.0	26.0	-	2022	mPLUG Effective & Efficient Language Learning
2	OFA	44.9	154.9	32.5	26.6	-	2022	OFA: Unifying Architectures, Task, & Modalities
3	GIT	44.1	151.1	32.2	26.3	-	2022	GIT: A Generative Image-to-text Transformer
4	BLIP-2 ViT-G OPT 2.7B	43.7	145.8	-	-	-	2023	BLIP-2: Bootstrapping Language-Image Pretraining
5	BLIP-2 ViT-G OPT 6.7B	43.5	145.2	-	-	-	2023	BLIP-2: Bootstrapping Language-Image Pretraining
6	ExpansionNet v2	42.7	143.7	30.6	24.7	61.1	2022	Exploiting Multiple Sequence Lengths in Fast end-to-end Training
7	LEMON	42.6	145.5	31.4	25.5	-	2022	Scaling Up Vision-Language Pre-training for Image Captioning
8	BLIP-2 ViT-G FlanT5 XL	42.4	144.5	-	-	-	2023	BLIP-2: Bootstrapping Language-Image Pretraining
9	GRIT (NO VL PRETRAINING)	42.4	144.2	30.6	24.3	60.7	2022	GRIT: Faster and Better Transformer Using Dual Visual Features
10	Prompt Tuning	41.81	141.4	31.51	24.42	-	2022	Prompt Tuning for Generative Multimodal Pretrained Models
11	OSCAR	41.7	140	30.6	24.5	-	2020	Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks
12	Xmodal-Ctx	41.4	139.9	30.4	24.4	-60.4	2022	Beyond a Pre-Trained Object Detector: Cross-Modal Textual and Visual Context

Figure 10: Current State-of-the-art Benchmarks

3 Image Captioning in Mongolian Language

3.1 Significance of Building Baseline Model

As mentioned in the previous sections, there are numerous technical difficulties in developing models for low-resource languages. It is crucial to understand the characteristics of the languages and it's performance against multi-modality with baseline models. Technical challenges include:

- **Limited Data Availability:** The most significant challenge in supervised learning is the scarcity of annotated datasets. Many languages lack large, diverse corpora necessary for training robust models. Creating datasets like MSCOCO for every language

is extremely challenging. The Flickr30K_MN dataset, created for this research paper, is the first of its kind in the Mongolian language.

- **Lack of Pretrained Models:** Training models from scratch is highly uncommon nowadays. For many languages, there are few or no pretrained models available, which are crucial for achieving satisfactory performance in NLP tasks.

- **Resource Disparity:** There is a disparity in the tools and resources (such as tokenizers, stemmers) available for low-resource languages compared to high-resource ones. For example, in English, the "spaCy" library [37] is commonly used for tokenization and text processing, which occurs automatically. In contrast, custom scripting methods were implemented for Mongolian in this research.

- **Cultural and Contextual Nuances:** Understanding cultural and contextual nuances is crucial for tasks like translation or sentiment analysis, which can be challenging for low-resource languages due to the limited context available. For instance, in quantitative analysis, common practices include using BLEU, CIDER, and METEOR scores. However, these evaluation metrics were initially developed in English, and there is considerable debate among researchers about their controversial use as universal metrics for all languages.

Henceforth, it is crucial to build the baseline models and achieve initial benchmark scores. The importance of building base models include:

- **Foundation for Advanced Models:** Base models provide a fundamental understanding of the language's structure and nuances. This knowledge is crucial for future improvement, accuracy, and efficiency.

- **Benchmarking:** Establishing base models sets benchmarks for the language, allowing researchers to measure and compare the performance of more advanced methods. It is the first time to achieve quantitative and qualitative evaluation in Mongolian language.

- **Dataset building:** Without data, it is impossible to train a model. By building dataset, it allows languages to be generalized and improve accessibility. In the age of multi-modal machine learning, the dataset usage is not limited to image understanding and can be utilized in other applications as well.

- **Community Engagement:** Starting with basic models can foster community involvement, crucial for languages with smaller speaker bases. Community contributions can help improve data collection and model refinement. For this inspiration, I will open source my dataset and source code for this research and share my benchmark results.

- **Transfer Learning:** Base models can serve as a starting point for transfer learning, where a model trained on one task is adapted for another. By uploading my model on platforms such as HuggingFace, other researchers can access the models weights and fine-tune or transfer learn for their specific tasks.

3.2 Overview of the Implemented Model

To the best of my understanding, this is the first implementation of an image captioning model in the Mongolian language. As previously discussed, establishing a baseline model and attaining preliminary benchmark scores is crucial for future development. Consequently, within the scope of this research, I have trained two distinct models: a basic encoder-decoder architecture and an advanced attention-based framework. The results are compared quantitatively and qualitatively, also the captions are compared to machine translated versions. I used PyTorch for my choice of implementation.

3.3 Model Architecture

3.3.1 Encoder: Image Features

For capturing image features, the model uses transfer learning. I used ResNet [42] which trained on IMAGENET1K_V1 dataset. These days, it is rare to train models for general computer vision tasks since there are many pre-trained models with excellent performance. Table [1], summarizes some of the famous pre-trained model. ResNet or Residual Neural Network is a type of CNN that skips connection to fir the input from the previous layer to the next layer without any modification. ResNet therefore can have a very deep network of layers up to 152-layers, by learning their residual functions instead of receiving the signals directly. IMAGENET is a dataset which contains over 15 million labeled high resolution images with over 22000 categories. The model is chosen based on the performance and ease of implementation with PyTorch framework.

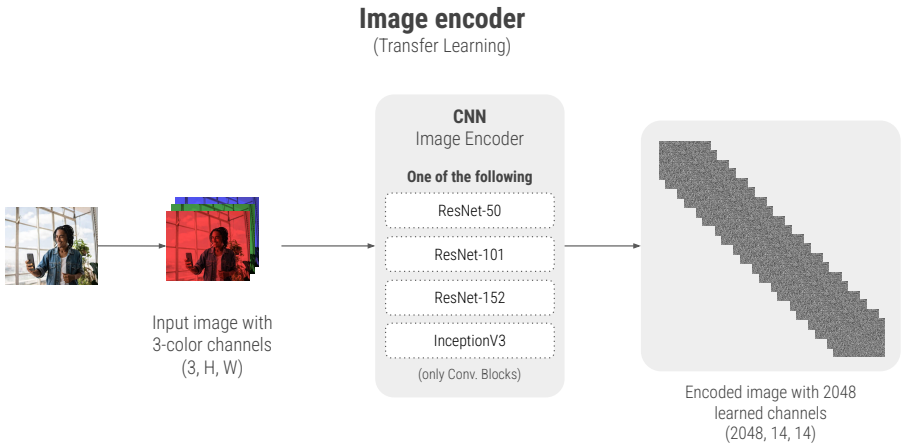


Figure 11: Image Encoder

The encoder is implemented in PyTorch framework as the following.

```
class Encoder(nn.Module):
    """
    Encoder.
    """

    def __init__(self, encoded_image_size=14):
```

```
        super(Encoder, self).__init__()
        self.enc_image_size = encoded_image_size

        resnet = torchvision.models.resnet152(weights=ResNet152_Weights.
    IMAGENET1K_V1)  # pretrained ImageNet ResNet-152

        # Remove linear and pool layers (since we're not doing
    classification)
        modules = list(resnet.children())[:-2]
        self.resnet = nn.Sequential(*modules)

        # Resize image to fixed size to allow input images of variable
    size
        self.adaptive_pool = nn.AdaptiveAvgPool2d((encoded_image_size,
    encoded_image_size))

        self.fine_tune()

    def forward(self, images):
        """
        Forward propagation.

        :param images: images, a tensor of dimensions (batch_size, 3,
    image_size, image_size)
        :return: encoded images
        """
        out = self.resnet(images)  # (batch_size, 2048, image_size/32,
    image_size/32)
        out = self.adaptive_pool(out)  # (batch_size, 2048,
    encoded_image_size, encoded_image_size)
        out = out.permute(0, 2, 3, 1)  # (batch_size, encoded_image_size,
    encoded_image_size, 2048)
        return out

    def fine_tune(self, fine_tune=True):
        """
        Allow or prevent the computation of gradients for convolutional
    blocks 2 through 4 of the encoder.

        :param fine_tune: Allow?
        """
        for p in self.resnet.parameters():
            p.requires_grad = False
        # If fine-tuning, only fine-tune convolutional blocks 2 through 4
        for c in list(self.resnet.children())[5:]:
            for p in c.parameters():
                p.requires_grad = fine_tune
```

Listing 1: Encoder Implementation

3.3.2 Decoder

For the decoder part, I have used Long Short Term Memory networks - or "LSTMs" [16] - which produces captions word by word conditioned on the context vector.

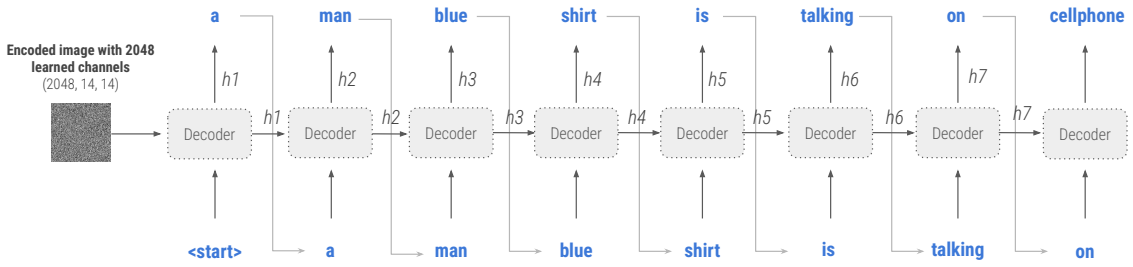


Figure 12: RNN-based Decoder

3.3.3 Attention Mechanism

Similar to human attention, which focuses on specific parts of the environment, while ignoring other parts, attention mechanism in AI allows models focus on specific parts of the input data. As result, contextually, it helps the model dynamically prioritize which parts of the data are most relevant to the task at hand.

For the base model, *Stochastic Soft Attention* is used, first implemented by Bahdanua et al., [41]. The attention layers is implemented same way as in the paper [28]. This network operates by applying seperate linear transformations to both the flattened encoded image $(N, 14x14, 2048)$ and the hidden stae output from the Decoder, aligning them to the same attention dimension. These transformed outputs are then combined and processed through a $ReLU$ activation. Subsequently, a third linear layer further refines this output to a single dimension, following which a softmax function is applied to generate the attention weights α.

- Encoder Attention: $A_e = W_e \cdot O$
- Decoder Attention: $A_d = W_d \cdot H$
- Combined Attention: $A = ReLU(A_e + A_d)$
- Attention Weights: $Softmax(W_a \cdot A)$

Where O is the output from the encoder, H is the hidden state of the decoder, W_e, W_d, W_a are trainable pparameters and α are the main attention weights.

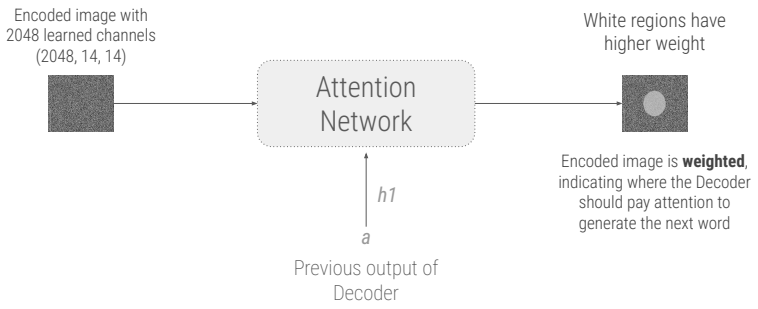


Figure 13: Attention Mechanism

Integrated Architecture

A integrated architecture can be seen from the Figure [14]. After the Encoder produces the encoded image, this encoding is transformed to initialize the LSTM Decoder's initial hidden state h, and cell state C. During each decoding step, the Attention network utilizes both the encoded image and the preceding hidden state to compute weights for each pixel. Henceforth, the LSTM Decoder receives both the previously generated word and a weighted average of the encoding to produce the subsequent word.

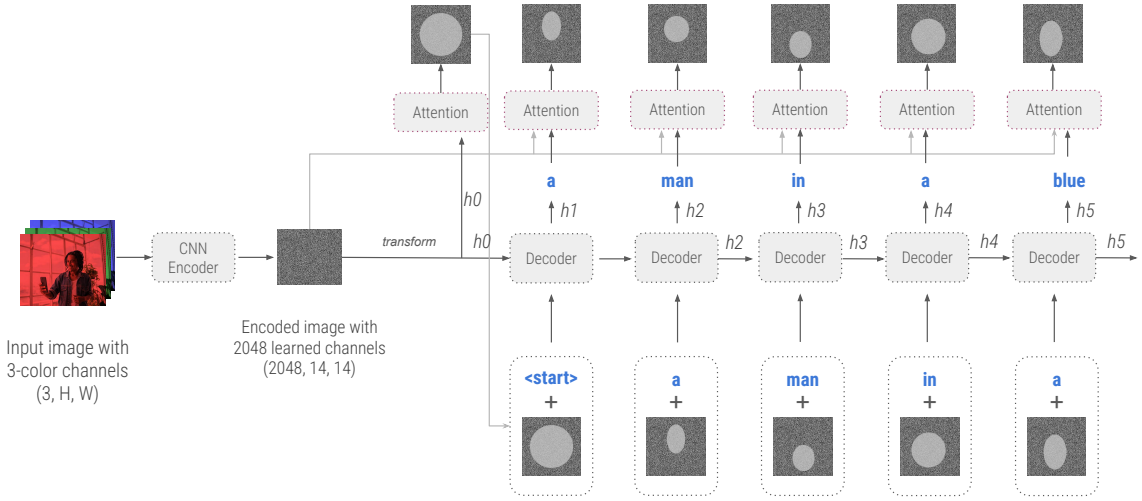


Figure 14: Combined architecture

3.3.4 Beam Search

The beam search algorithm is a widely used technique in various generative AI tasks, including text summarization, machine translation, image captioning, and automatic speech recognition. It operates by employing a linear layer to transform the Decoder's output into scores for each word within the vocabulary and it is a variant of breadth-first-search algorithm. While an alternative algorithm known as greedy search exists, it functions by selecting the highest-scoring word and utilizing it for subsequent word prediction. This approach can prove suboptimal in NLP tasks like caption generation, as the remainder of the sequence becomes overly reliant on the initial word choice. In contrast, beam search defers

the determination of sequence order until the decoding process is fully complete. It then selects the sequence that boasts the highest overall score. Beam search algorithm works as the following:

- Start with an image that has been processed through a model (CNN-LSTM for this case) to generate a probability distribution over starting words of caption. Then initialize the beam search width width B, which is the number of sequences to be considered at each step. (If $B = 3$, the algorithm will keep the top 3 most probable sequences at each step.)
- Once the top-B most probable words are selected, those from the initial B paths in the search tree.
- Expand each of the B paths by one additional word, creating several new candidates.
- Once the beam search is complete, the sequence with the highest cumulative probability is typically chosen as the output caption.

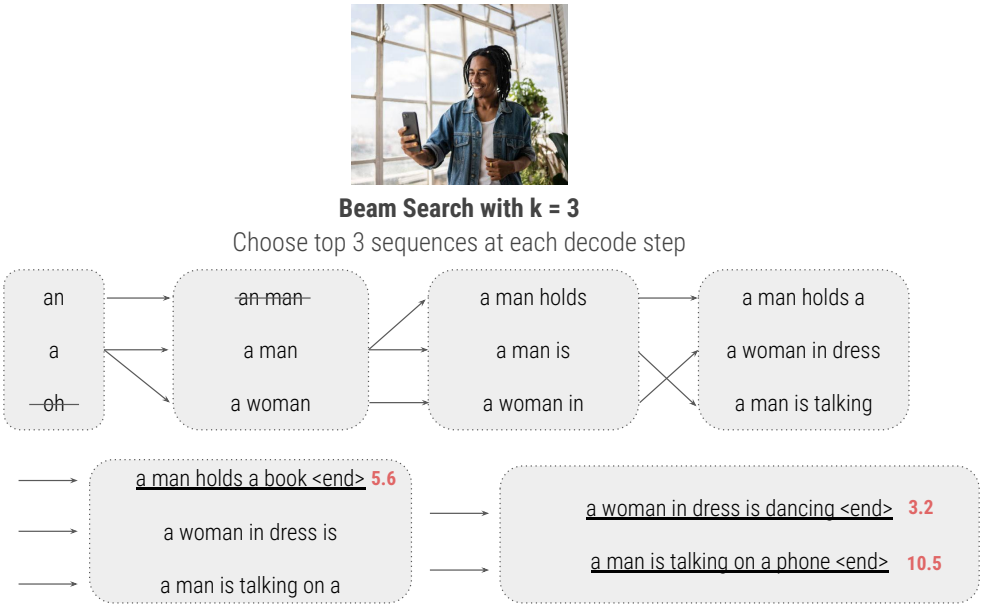


Figure 15: Beam Search Algorithm Example: When $B = 3$

3.4 Dataset Preparation

The Flickr30K dataset is widely used resource in the field of computer vision and natural language processing, particularly for tasks involving image captioning and visual questions answering. This dataset consists of 31,783 images collected from Flickr, along with five captions for each image. These captions have been written by real human annotators, providing diverse and descriptive text for each image.

The original source of the Flickr30K dataset is a paper by Bryan A. Plummer et.al [26]. Their work presents the dataset and discusses its utility in image captioning and other tasks

that combine image and text processing. The Flickr30K dataset is intended primarly for non-commercial research and educational purposes. The creation of Flickr30K_MN dataset also adhere the same terms of use stated by Flickr, focusing on non-commercial, research, and educational purposes.

3.4.1 Development of dataset creating tool

The dataset creation tool uses streamlit for the front-end and Firebase's Firestore as the database. With real-time tracking of the translated data and user authentication, this allows multiple translators to contribute to the project at the same time.

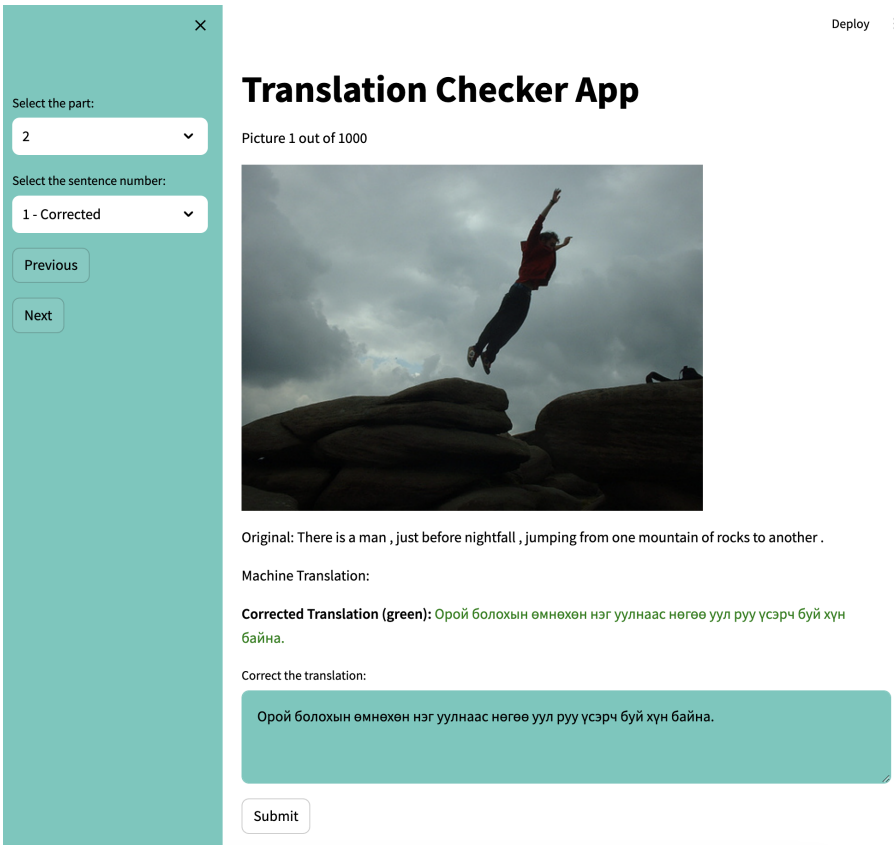


Figure 16: Annotation tool developed for this Research

3.4.2 Dataset Creation Process

In this IRP, Flickr30k dataset is used. In order to create high quality dataset in limited amount of time and human resource, an automation process was necessary. The dataset was first translated by Google Translate's opensource API [27] and then post edited by native Mongolian translators also fluent in English. Front-end is used to automate the process in the following manner:

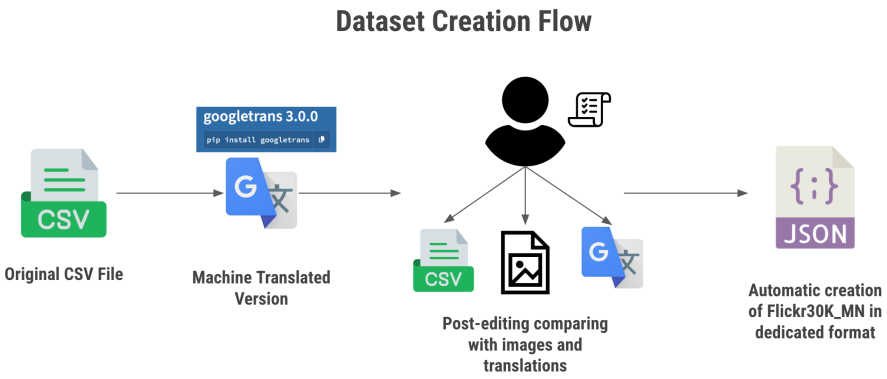


Figure 17: Flickr30K_MN Dataset creation workflow

The final format of the dataset in JSON format looks like this:


```
{
"images": [
    {
        "sentids": [0, 1, 2, 3, 4],
        "sentences": [
            {
                "tokens": [
                    "хоёр",
                    "залуу",
                    "хашааны",
                    "дэргэд",
                    "зогсож",
                    "байна"
                ],
                "raw": "хоёр залуу хашааны дэргэд зогсож байна",
                "imgid": 1000092795,
                "sentid": 0
            },
            {
                "tokens": [
                    "хоёр",
                    "залуу",
                    "юм",
                    "ярилцан",
                    "гадаа",
                    "зогсож",
                    "байна"
                ],
                "raw": "хоёр залуу юм ярилцан гадаа зогсож байна",
                "imgid": 1000092795,
```

```
            "sentid": 1
        },
```

3.5 Dataset Preprocessing

The `create_input_files` function in the `utils.py` designed to process image datasets and segment them into TRAINING, VALIDATION, and TEST sets. TRAINING set include 29783 images, whereas VALIDATION and TEST sets both include 1000 images each.

3.5.1 Word Frequency Count and Word Map Creation

The function counts the frequency of each word across all captions to identify common words. Based on this count and a minimum frequency threshold, it creates a "word-map" which is dictionary mapping words to unique integer indices. Special tokens for <unknown>, <start>, <end>, and <padding> are also added to this map.

3.5.2 Image and Caption Processing

For each image, the function processes and resizes to a consistent size (256x256 pixels) and pairs it with a fixed number of captions (<`captios_per_image`>.) Each caption is truncated or padded to a maximum length.

3.5.3 Encoding and Saving Data

The captions are encided using the word map, and the images are saved in an HDF5 file. Encoded captions and their lengths are saved in seperate JSON files.

3.6 Training

These are the hyperparameters for the training.

```
# Model parameters
emb_dim = 512  # dimension of word embeddings
attention_dim = 512  # dimension of attention linear layers
decoder_dim = 512  # dimension of decoder RNN
dropout = 0.5
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
cudnn.benchmark = True

# Training parameters
start_epoch = 0
epochs = 120  # number of epochs to train for
epochs_since_improvement = 0  # keeps track of number of epochs
batch_size = 80
workers = 1  # for data-loading; right now, only 1 works with h5py
encoder_lr = 1e-5  # learning rate for encoder if fine-tuning
decoder_lr = 1e-5  # learning rate for decoder
grad_clip = 5.  # clip gradients at an absolute value of
alpha_c = 1.  # regularization parameter for 'doubly stochastic attention
    ', as in the paper
```

```
best_bleu4 = 0.  # BLEU-4 score right now
print_freq = 100  # print training/validation stats every __ batches
fine_tune_encoder = False # fine-tune encoder? or True
checkpoint = # path to checkpoint, None if none
```

Listing 2: Hyperparameters for the training

In the training process, first only the Decoder is trained for the initial 12-13 epochs where the BLEU4 score averages 15.0 (it was observed the loss and accuracy do not improve substantially once they reach epochs 12-15). Once they reach around the BLEU4 score 15.0, the fine-tuning is performed on the encoder to further improve the training process. All models follow a similar training procedure for different encoder components.

3.7 Loss Function

The loss function used in this implementation is CrossEntropyLoss, which is commonly applied in sequence generation tasks such like word sequence prediction. A raw scores of the final Decoder layer is provided and the CrossEntropyLoss internally applies a softmax function to these scores to obtain a probability distribution, and then calculates the logarithm of these probabilities to compute the loss.

Mathematically, if $p(y|x)$ is the probability distribution output by the softmax function for the true class y given the input, the CrossEntropyLoss L can be expressed as:

$$L == \sum_{i=1}^{N} \log p(y_i|x_i) \tag{12}$$

where N is the number of training samples, y_i is the true label for the i-th sample, and x_i is the i-th input. Same approach as in the paper [28], the authors suggest emplying "doubly stochastic regularization." The regularization ensures that the attention weights α not only sum to 1 across all pixels at each timestep but also encourage pixel p to receive a cumulative weight of 1 across all timestep T. The regularization term can be expressed as:

$$R = \sum_{p=1}^{P} \left(1 - \sum_{t=1}^{T} \alpha_{tp} \right)^2 \tag{13}$$

where P is the number of pixels, T is the number of timesteps, and α_{tp} is the attention weight for pixel p at the timestep t. The overall loss function becomes a combination of the CrossEntropyLoss and this regularization term.

3.8 Computational Resource

The model is trained on the following computational resource.

- CPU: Intel Core i9-10940X @ 3.30GHz
- Memory: 64GB
- GPU: Nvidia GeForce RTX 3090 24GB * 2
- OS: Ubuntu 20.04.2 LTS

4 Results

4.1 Quantitative Results

For quantitative results, the model performance has been evaluated on the BLEU(1,2,3,4), METEOR, and CIDEr metrics. Additionally, the model performance is compared with machine translation on the test set. Summary of results can be seen from the Figure [18].

Quantitative Results

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
(base) ResNet50	0.50	0.31	0.19	0.11	0.17	0.35
(base) ResNet101	0.52	0.33	0.21	0.13	0.18	0.40
(base) ResNet152	0.54	0.35	0.23	0.15	0.19	0.44
(attn) ResNet50	0.59	0.42	0.28	0.19	0.20	0.40
(attn) ResNet101	0.61	0.43	0.30	0.20	0.20	0.40
(attn) ResNet152	0.62 ↑ 24%	0.44 ↑ 41%	0.31 ↑ 63%	0.21 ↑ 90%	0.32 ↑ 88%	0.46 ↑ 31%
(MT) Google Trans	0.49 ↓ 2%	0.34 ↑ 9%	0.23 ↑ 21%	0.17 ↑ 54%	0.45 ↑ 164 %	0.0 unk

Figure 18: Quantitative Results: Comparison between baseline model, attention-based model, and machine translation

Baseline model trained on ResNet as backbone

The base models with ResNet backbones show a progressive improvement as the depth of the ResNet architecture increases (from ResNet50 to ResNet152). This suggests that deeper networks have captured more complex features from the images that are beneficial for the captioning task. The METEOR and CIDEr scores improve with the depth of the network, indicating better quality captions in terms of both semantic content (METEOR) and n-gram matching with human judgment (CIDEr).

Attention-based model trained on ResNet as backbone

The attention-based ResNet models outperform their base counterparts across all metrics. This indicates that the attention mechanism helps the model focus on the most relevant parts of the image when generating a caption, leading to more accurate and coherent captions. ResNet152 with attention achieves the highest scores across almost all metrics, with particularly notable improvements in CIDEr and METEOR, suggesting that it generates captions that are well-aligned with human judgments and reference captions.

4.1.1 Discussion on the Quantitative Result

The main goal of the quantitative evaluation was to achieve the baseline score and observe how do the metrics improve through modifications. To realize the optimal training strategy, multiple trials and errors were conducted to attain the best performance. Once the highest performance was achieved, other models were trained in a similar manner. The highest overall score was achieved with the attention model that used ResNet152, and LSTM as the backbone. The training strategy is as follows:

- Train the decoder until the BLEU-4 score peaks at certain point. (Resnet152, attention peaked at 15th epoch)
- The initial learning rate was set to 3e-4. The learning rate decay is applied after 8 epochs of no improvement, and the training process will be terminated if the BLEU-4 score does not improve after 20 consecutive epochs.
- Once the BLEU-4 score peaks during Decoder-only training, fine-tuning on the ResNet will start, further training the model.
- Terminate once BLEU-4 score peaks at a certain epoch.

Based on the performance, following discussions can be made:

Computation and Dataset Bottleneck

One observation from the training was the occurrence of overfitting around epochs 25-30. In the case of the attention-based ResNet152, the model showed no improvement after epoch 30. Similar patterns were also observed in the other models. This indicates that with this approach, the dataset size becomes a bottleneck, and the model ceases to improve after a certain epoch. To achieve a satisfactory score, this approach has limitations, and a different approach is necessary. Although the Flickr30K dataset is not small, it falls short compared to larger datasets such as COCO. One possible way to overcome this difficulty is to use both Translation-based and Alignment-based approaches in the model architecture. The paper [19] proposes using a pre-trained English caption model to generate high-quality English captions, and then use both the image and generated English captions to generate captions in low-resource languages. This method, which also utilized the Flickr30K dataset, can improve the performance of low-resource language image captioning capabilities in other languages as well. In future work, more advanced architectures will be implemented, including a transformer decoder, vision-transformer encoder, and unsupervised methods.

Language-specific Characteristics and Limitations of Quantitative Metrics

Although achieved BLEU4, METEOR, and CIDEr scores in this research are no near to the current state-of-the-art scores in English standards, it is important to handle these metrics with great care. For starters, BLEU, METEOR, and CIDEr were primarily developed for English and the highest BLEU4 score trained on the Flickr30K dataset is 30.1, whereas the highest BLEU4 score on the Mongolian language in this research is 20.1. This gap in performance is either because of the fact that model implemented in this work is inferior to more recent models or the quantitative metrics are not capturing the linguistic nuances of Mongolian language. An example of why BLEU is less suitable in a multilingual setting can be seen from the German language, as it is well known for its poor performance on

these metrics. German has flexible word order, extensive compound word formation, and three grammatical genders. Additionally, the length of German sentences tends to be longer than English sentences. Hence, it is crucial to consider language specificity in this kind of evaluation. Related research papers often focus on quantitative scores and neglect linguistic features and qualitative analysis. Further discussion and research are necessary for the generalization of tasks related to low-resource languages. An interesting method to consider is including the linguistic characteristics in the feature space and using that in the training process.

Vocabulary Size Dependency on the Training Data

The vocabulary was created based on the created dataset on the condition that a new token would be added only if it appeared more than five times. When there is limited training data, the vocabulary size is constrained by the input, which could lead to poor alignment between the complex image features captured by the pre-trained model and the amount of vocabulary available to describe them in words. In future work, it is worth exploring the use of pre-trained word embeddings, such as BERT word embeddings, which include 30,000 words.

Machine Translation and Image Captioning Comparison

As the depth of the CNN model increased, both the base model and the attention model outperformed machine-translated sentences on the TEST set. It is believed that the poor quality of the translation led to this performance difference. The original dataset was first machine-translated and then post-edited by native Mongolian speakers. It was observed that Google Translate tends to output words in English if there is no equivalent word in Mongolian. Additionally, Google Translate often produces arbitrary repeated words without capturing the meaning of the sentence. In comparison, the captions generated by the image captioning model are based on the alignment of image features with an attention mechanism that focuses on relevant parts of the image. This characteristic promotes the validity of multi-modal machine translation.

4.2 Qualitative Results

Qualitative analysis is conducted on previously unseen image samples. This analysis aims to determine if the model accurately focuses on the relevant regions of the image during the sequential generation of words. Total of 50 randomly selected images from the test set were analyzed. Out of 50 images, 6 were categorized as "insufficient" whereas rest of the 44 were "satisfactory". The evaluation is based on the following criteria:

- **Relevance:** Can the model generate contextually accurate sentences?
- **Accuracy:** Can the model generate semantically and grammatically correct sentences?
- **Attention:** Does the model focus on the correct region of the image when generating each tokens?
- **(Additional:)** Can the model identify cultural-specific or region-specific objects or actions in the image?

Sufficient Example 1: Beam Search=3

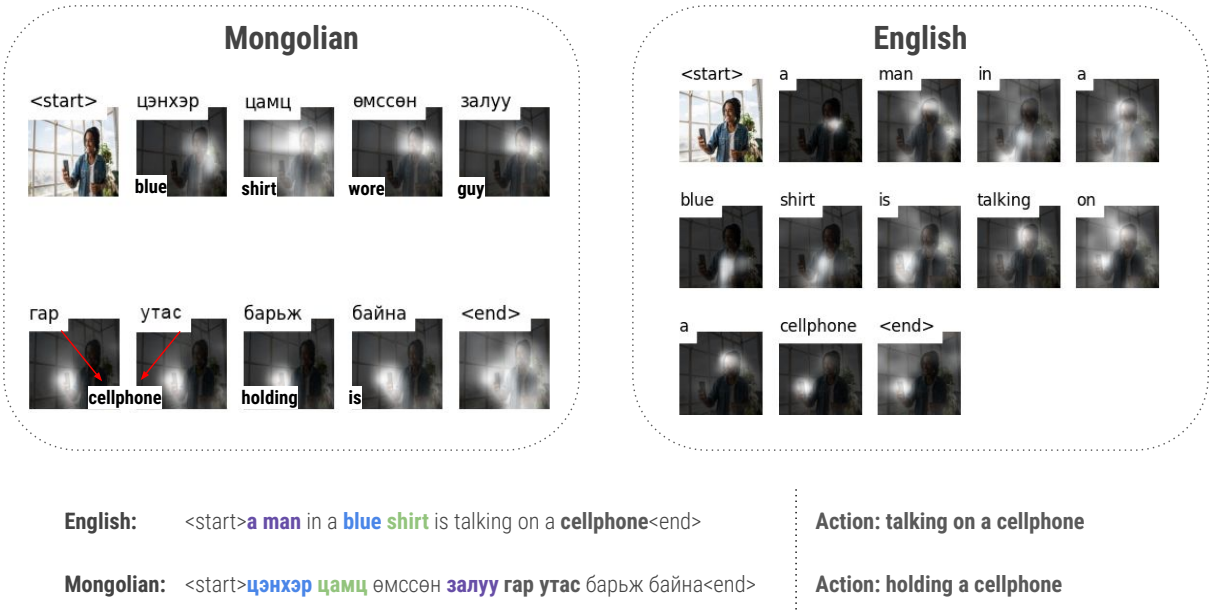


Figure 19: Qualitative Evaluation: Example 1 (Beam size = 3)

In the first example, the model correctly identifies a man wearing a blue shirt and holding a cellphone. One interesting observation in the two results is that the action is described as "talking on a cellphone" in English, but as "holding a cellphone" in Mongolian. The attention weights are visible in the pictures, highlighting the correct regions of the image. For instance, there are highlights in the correct regions associated with the shirt and cellphone.

Sufficient Example 2: Beam Search=3

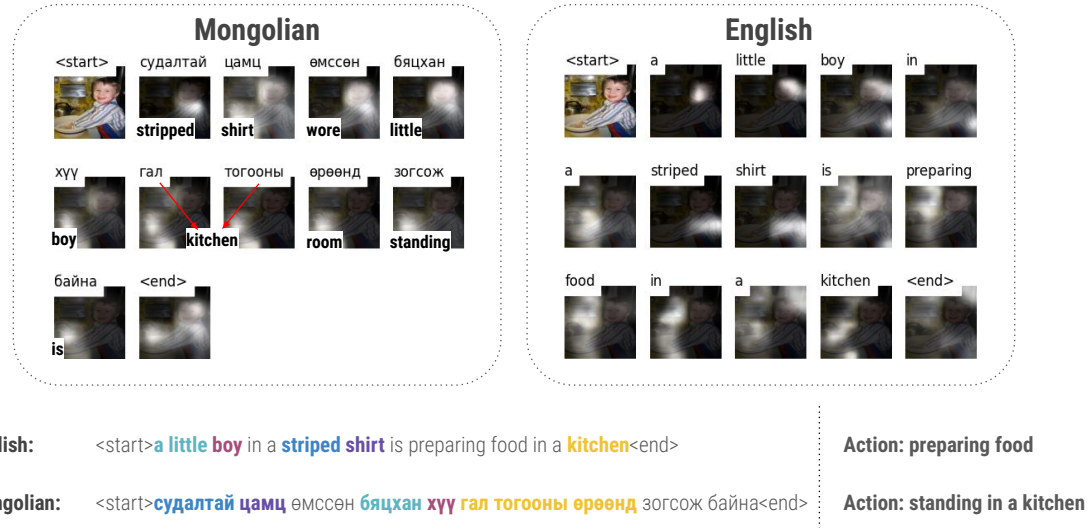


Figure 20: Qualitative Evaluation: Example 2 (Beam size = 3)

In the second example, the model correctly identifies that the little boy is wearing a striped shirt. Additionally, the model accurately describes the surroundings, which include a kitchen and the preparation of food.

Sufficient Example 3: Beam Search=3

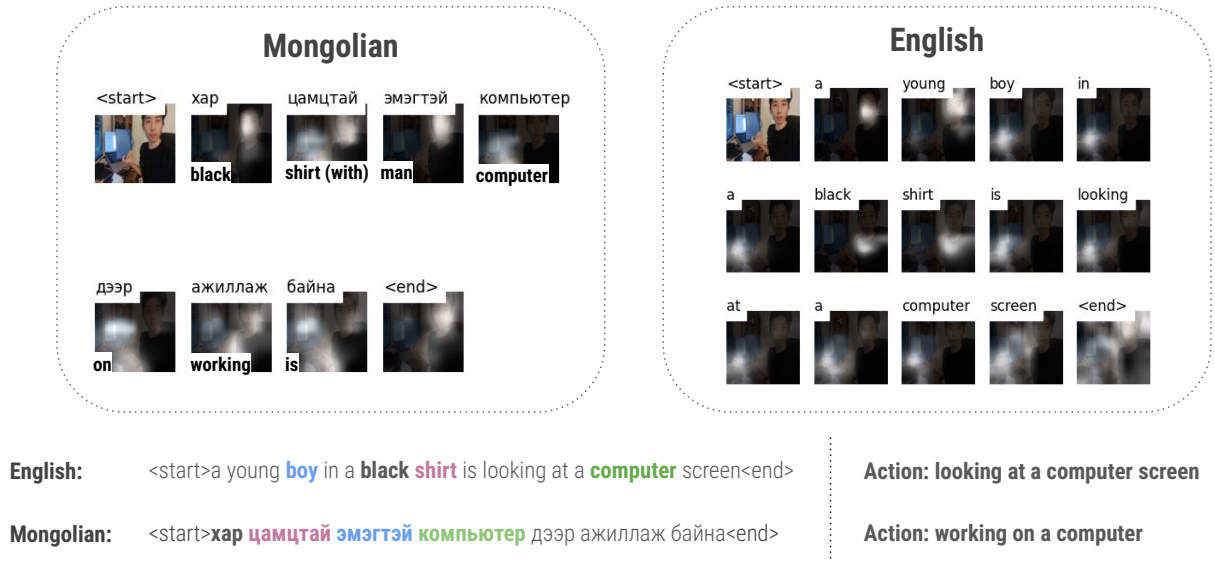


Figure 21: Qualitative Evaluation: Example 3 (Beam size = 3)

In the third example, the model correctly identifies that a man in a black shirt is working on a computer. The model trained on the English dataset states that the boy is looking at a computer screen, whereas the model trained with the Mongolian dataset outputs that the man is working on a computer. The model accurately highlights the correct regions of the image: black shirt, man, and computer. However, the subject is also holding a book, which is not mentioned in either of the descriptions.

Sufficient Example 4: Beam Search=3



English: <start>**a woman in a blue shirt** is **talking to a man** in a blue shirt<end>

Action: talking to a man

Mongolian: <start>**цэнхэр цамц** өмссөн **эмэгтэй гал тогооны өрөөнд зогсож** байна<end>

Action: standing in the kitchen

Figure 22: Qualitative Evaluation: Example 4 (Beam size = 3)

In the fourth example, the model correctly identifies a woman wearing a blue shirt standing in the kitchen. The model trained on the English dataset describes both subjects but does not mention the kitchen. In contrast, the model trained on the Mongolian dataset recognizes that the woman is in the kitchen but does not mention the man. Additionally, the model accurately recognizes and highlights the correct features such as blue, woman, and kitchen.

Sufficient Example 6: Beam Search=3



English: <start>**a man in a black leather jacket and a black hat is reading a newspaper** <end>

Action: reading a newspaper

Mongolian: <start>**хар хүрэмтэй хүн галт тэрэгний буудал дээр сонин уншиж байна**<end>

Action: reading newspaper at the station

Figure 23: Qualitative Evaluation: Example 6 (Beam size = 3)

In the sixth example, the model outputs that there is a man wearing a black jacket reading a newspaper at the station. Regarding the model trained on the English dataset, it describes the subject but does not mention that he is at the station. In both cases, the models output that the man is reading a newspaper, whereas in reality, he is reading a book. The model accurately highlights the relevant parts of the image such as newspaper, station, and black jacket.

Sufficient Example 7: Beam Search=3

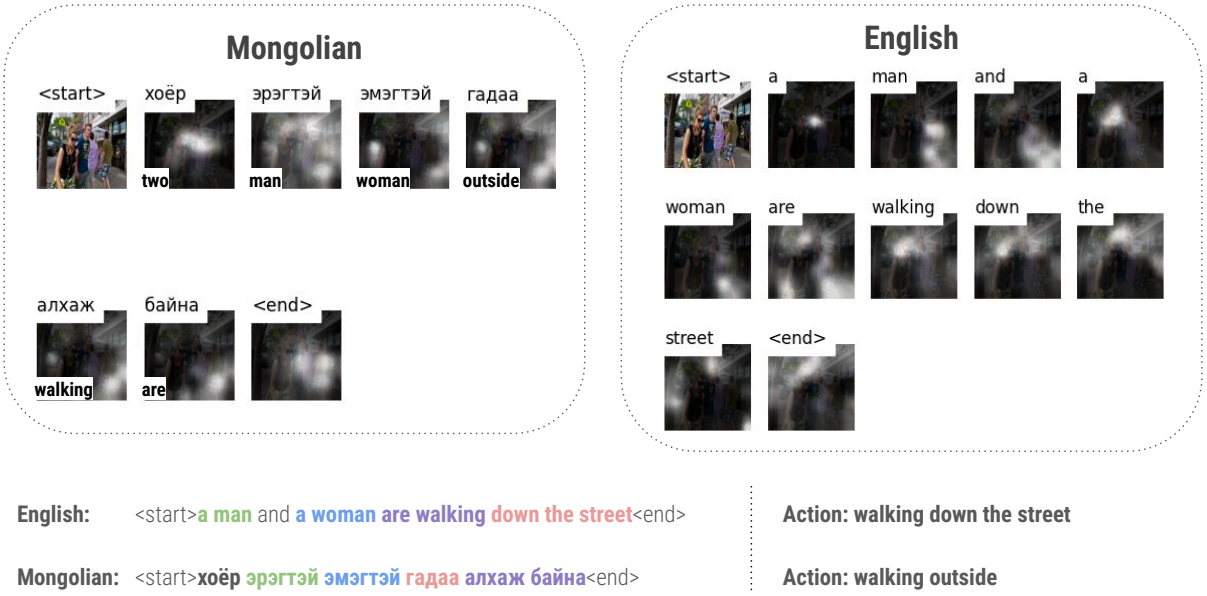


Figure 24: Qualitative Evaluation: Example 7 (Beam size = 3)

The model states that there is a woman and a man walking outside. In the English version, the model outputs that there is a man and a woman walking down the street. The difference in action lies in the phrases "walking down the street" and "walking outside".

Poor Example 1: Beam Search=3 In this example, the model incorrectly perceives a man and a woman in the picture, whereas in reality, there is only one man who is talking on the phone and holding a bag. The model could not infer that the man is holding a phone, as it is not visible in the picture. Additionally, it suggests that the man is sitting on the road, whereas, in reality, he is walking quickly.



Figure 25: Insufficient Example: Mirror Reflection

It can be said that the model struggles with complex scenes like this one, where reflections or the main action or objects are hard to distinguish from visual features.

Poor Example 2: Beam Search=3 This is a poor example because it essentially fails to meet any of the criteria. First, the model cannot generate contextually accurate descriptions. For instance, the output states that the man is wearing a grey shirt and black shorts, when in fact, he is shirtless and wearing grey shorts. Additionally, it is evident from the picture highlight that the model is not focusing on important parts of the image. For example, the model did not concentrate on the dog, and as a result, the dog is not generated. We can also observe that it is focusing on the bench, which is not particularly relevant in this example.

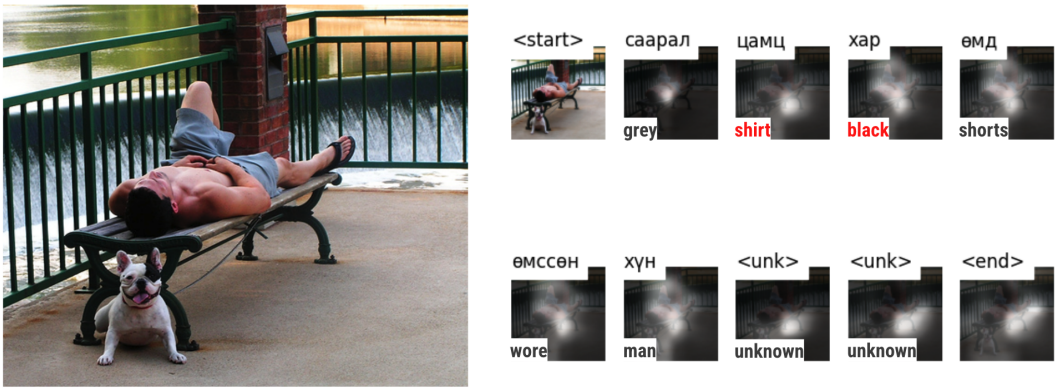


Figure 26: Poor Example: Not Attending relevant part of the image

Poor Example 3: Beam Search=3

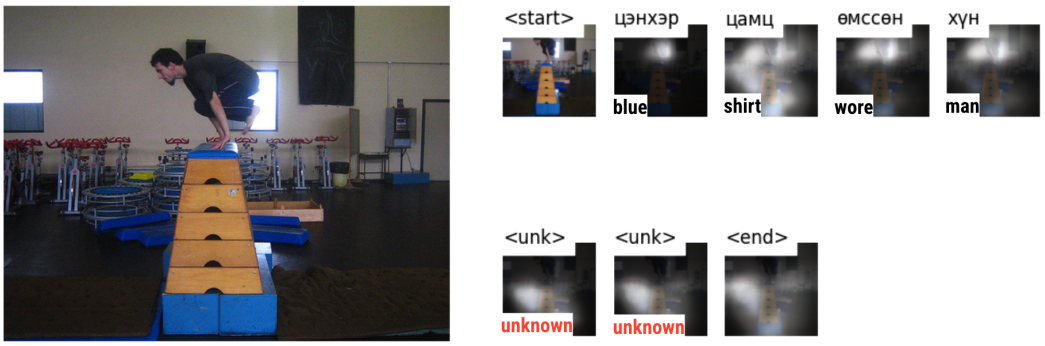


Figure 27: Poor Example: Not recognizing objects

In this case, the model failed to identify the gymnastic vaulting box, a term specific to gymnastics. It is evident from the highlighted section that despite focusing on the vaulting box, there was a lack of proper alignment with the term. This alignment issue can be improved by incorporating more datasets relevant to this context. Furthermore, the model neglected to describe the jumping action shown in the image, which is a crucial element of the picture.

Poor Example 4: Beam Search=3



Figure 28: Poor Example 4: Poor alignment in image and text

In this example, the model fails to detect the Lego toy depicted in the image. Similar to Figure [27], the model's attention is on the Lego house, yet it does not associate a corresponding word with it. However, the model correctly recognizes that a little kid is playing with something, but it does not identify what the child is playing with.

Additional Test: Cultural-specific sample 1



Figure 29: Cultural specific: Mongolian wrestling

In this example, we see two Mongolian athletes wearing special outfits and wrestling. Given the lack of Mongolian-specific content in the dataset, it is unlikely that the model would specifically mention that the athletes are wrestling in Mongolian outfits. Instead, it inaccurately states that there are two men running.

Additional Test: Cultural-specific sample 2



Figure 30: Cultural specific: Mongolian Greeting

In the next example, we observe two men dressed in traditional Mongolian attire greeting each other. If human annotators were to describe this picture, they would likely mention the type of outfits being worn and the manner of their greeting. The generated sentence states that two men wearing blue outfits are sitting together, which is somewhat accurate but fails to capture the important context.

Additional Test: Cultural-specific sample 3

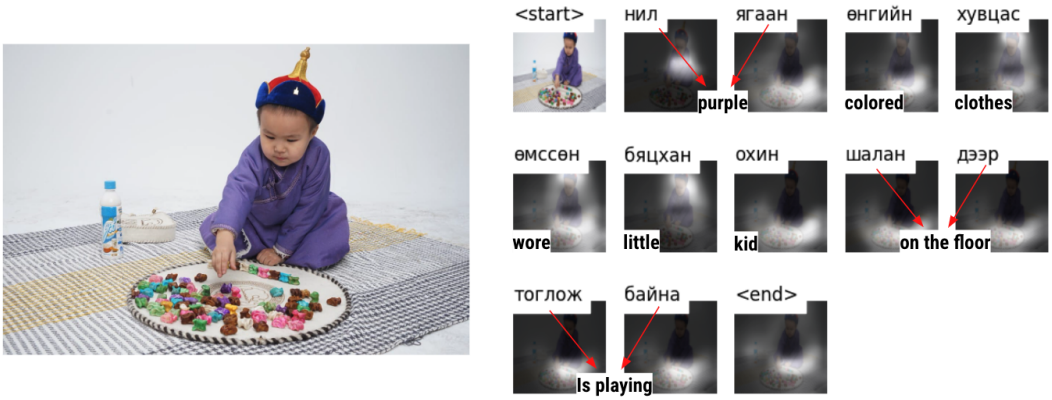


Figure 31: Cultural specific: Mongolian Game

In the next example, we see a kid wearing traditional clothes playing with "*shagai*," a traditional Mongolian game made from sheep or goat ankles. The generated output describes a kid wearing a purple outfit playing on the floor. While this caption is accurate, it fails to mention what the kid is playing with and does not explain the context.

4.2.1 Conclusion

A total of 50 randomly selected images from the test set were used for qualitative analysis based on the evaluation criteria. Out of these 50 images, 6 were found to be insufficient in terms of performance, due to issues such as poor alignment of image features with words, not focusing on relevant parts of the image, and contextually and grammatically incorrect descriptions. In contrast, sufficient examples were able to generate contextually accurate and grammatically correct sentences, while focusing on the relevant parts of the image. These sufficient examples were also compared with English captioning. An interesting observation was that the Mongolian model tended to explain the surroundings in more detail in addition to the subjects. In comparison, the English model tended to focus on the subjects within the picture and mentioned less about the surroundings. This is fundamentally different from machine translation where the output is fully dependent on the input text. Whereas in image captioning, it decides on its own which part of the image to focus on and what to explain. This resembles human attention and can be useful for a wide range of applications including supporting tools for visually impaired individuals.

Additional tests were conducted to determine if the model can recognize culturally specific features within images. Figures [29], [30], and [31] feature Mongolian-specific clothing, actions, and objects. While the model can describe what is happening in the images, it lacks the vocabulary to state the correct terms, and therefore, cannot generate culturally specific terms. This type of insufficiency can be addressed through fine-tuning with compact and specific datasets.

In conclusion, from a qualitative perspective, the model demonstrated good performance, with only 1 out of 10 samples considered insufficient. It can provide basic context about what is happening inside picture. The model performed poorly on complex scenes such as overlapping objects or reflections on various surfaces. Additionally, the model struggled with

culturally specific contexts. If the model were to be used for very specific use cases, it might be reasonable to train a model on a smaller scale using a specific dataset, or to use language models that possess a substantial amount of knowledge about Mongolian (not limited to Mongolian), and connect them to a vision model.

5 Discussion and Conclusion

5.1 Thesis Contributions

In this Independent Research Project, the contributions are the following:

- Created an extension of Flickr30K_MN in Mongolian version. Current dataset can express the content of an image to a certain degree but more high-quality, annotated dataset is necessary for further research.
- Achieved baseline model and initial benchmark scores, which is crucial for future development. Intriguing finding was that when same dataset was trained on the English dataset, the quantitative scores were lower on the Mongolian dataset. This suggests that the current evaluation metrics might under perform on Mongolian language. Similar assumptions can be made on the Altaic region languages.
- Trained an attention-based model, which showed an improvement over the baseline model. However, with this approach, the dataset will continue to be a bottleneck.
- Results showed that the model struggled with generating culturally-specific scenarios including contextual nuances. Capturing cultural contexts like traditional games or ceremonies, requires linguistic understanding but also cultural knowledge.

5.2 Future work

In this research, a simple encoder-decoder model was implemented, and to improve performance, an attention-based model was developed. The attention-based model outperformed the baseline model but could not improve after certain stage. For future work, more advanced models, including the Vision Transformer (which will replace CNN), will be explored. Additionally, an approach that incorporates linguistic and culture-specific features into the training process will be examined. Another approach could be applied to areas where the model underperforms. For example, methods such as reinforcement learning through human feedback could enhance performance without dramatically scaling up the model. To further validate the results obtained from this research, exploring languages closely related to Mongolian would be beneficial.

5.3 Conclusion

In this research, we implemented the first Mongolian image captioning model. We trained two different types of models and established an initial benchmark for future reference. The trained models surpassed machine-translated sentences, suggesting that multi-modal models

can perform better than text-based machine translation. This improved performance is likely due to the fact that images provide additional context to the sequence generation process, and attention mechanism mimics human attention, outputting significant part of the image.

Let me revisit the research questions and answer them based on the research findings:

- What are the specific challenges in developing an image captioning model for the low-resource language?
  Answer: The scarcity of large, annotated datasets for Mongolian is crucial. Although Flickr_30K dataset is not small, it is not enough for capturing unique aspects of the language and visual information. For the next development, it is necessary to prepare dataset on Mongolian-specific images.

- What linguistic aspects of the Mongolian language are most challenging to capture in an image captioning model?
  Answer: The BLEU4, METEOR, and CIDEr scores obtained in the research are considerably lower compared to English standards. This discrepancy may be due to the model's limitations or the metrics' inability to capture the linguistic nuances of the Mongolian language. Several examples during analysis illustrate that the structure and morphology of Mongolian can differ significantly from English. The agglutinative nature, the tendency to omit certain verbs, and the difference in detail level can cause the standard metrics, which rely heavily on exact word matches and order, to misrepresent the quality of a translation or caption. This is why BLEU, METEOR, and CIDEr scores may not be fully compatible with Mongolian, as they fail to account for these linguistic features and the context-dependent nature of language.

- How can insights from developing this model inform linguistic research, especially regarding Mongolian and similar languages?
  Answer: The model architecture employed in this research utilizes well-studied generalization techniques including dropout, batch normalization, early stopping, and hyper-parameter optimization. In theory, the model should be able to capture complex characteristics with a reasonable amount of data. However, since the model's weights depend on the prepared dataset, it could not effectively learn about culturally specific features. This suggests that high-quality captions cannot be generated without possessing the knowledge in the first place. Perhaps, an reasonable approach would be using a model that possesses vast amount of knowledge such as English and incorporate language-specific translations into the target languages.

- What are the potential areas for further research in improving image captioning?
  Answer: From the perspective of low-resource languages and generalization, capturing linguistic features from these languages, similar to attention mechanisms, might be an interesting approach to incorporate into the training process as one of the resources. More complex embedding mechanisms or human-assisted machine translation techniques, such as reinforcement learning through human feedback, could help bridge some of the discrepancies. Additionally, with the advent of very powerful Large Lan-

guage Models (LLMs), more compact and reasonably sized task-specific models can now also be developed and shows promising prospectives.

Additional Remarks:

Quantitatively, the model's performance improved as its depth and complexity increased. However, all models reached a saturation point around epoch 25, indicating that the dataset remains a bottleneck for this encoder-decoder approach. Similar trends are likely to appear in the Altaic language family. If the goal is to increase performance while limiting the amount of training data, alternative methods such as unsupervised learning or a deep learning approach coupled with reinforcement learning could be promising approach. However, these are hypotheses for further research and to validate them, it would be necessary to observe other languages closely related to Mongolian. Additionally, although BLEU, METEOR, and CIDEr scores are primarily designed for English, there is a possibility that they may underperform with these languages. Based on these findings, if similar trends appear in other closely related languages, an alternative evaluation metric could be proposed.

Qualitatively, the model learned to focus on relevant parts of an image in generating captions, as evidenced by the alignment of image features with words. Nonetheless, it was observed that the model struggles with complex scenarios, such as when subjects hold multiple objects, when objects overlap, or reflections. Furthermore, we tested the model with a few examples to determine if it could identify culturally-specific content. While the model could explain the context to a certain extent, it failed to generate precise terms in cultural-specific context. These limitations could be addressed through fine-tuning with specialized datasets.

For practical applications, the image captioning model could be optimized for visually impaired individuals, for instance, by connecting it to a camera and voice assistance system. Alternatively, such a model can be used for managing large sets of unstructured data to extract insights from them.

To conclude, while the performance of the models improved with increased depth and complexity, all models stopped improving after certain epoch. Further research, possibly exploring unsupervised learning or a combination of deep learning and reinforcement learning, could enhance performance, especially with limited training data. The need for such research is required by the limitations of current evaluation metrics like BLEU, METEOR, and CIDEr when applied to languages other than English, suggesting the potential development of alternative metrics for more accurate assessment. Future research will involve more advanced architectures and methods that could take into account linguistic features.

References

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. *Show and Tell: A Neural Image Caption Generator*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3156-3164. https://api.semanticscholar.org/CorpusID:1169492.

[2] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 60(2):91–110, Nov. 2004. ISSN 1573-1405. doi:10.1023/B:VISI.0000029664.99615.94. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[3] T. Ojala, M. Pietikäinen, and T. Mäenpää. *Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns*. In Computer Vision - ECCV 2000, pages 404–420, Berlin, Heidelberg, 2000. Springer. ISBN 978-3-540-45054-2. doi:10.1007/3-540-45054-8_27. https://doi.org/10.1007/3-540-45054-8_27.

[4] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga. *A Comprehensive Survey of Deep Learning for Image Captioning*. ACM Computing Surveys, 51(6), Feb. 2019. ISSN 0360-0300. doi:10.1145/3295748. https://doi.org/10.1145/3295748.

[5] Sebastian Raschka, Vahid Mirjalili, *Python Machine Learning - Third Edition*, Packt Publishing, December 2019, ISBN 9781789955750.

[6] Sumit Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, Published in Towards Data Science, December 16, 2018, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53,

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, 2012.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] C. Szegedy et al., "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[11] C. Szegedy et al., "Rethinking the Inception architecture for computer vision," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[13] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[14] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[15] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur, *Extensions of Recurrent Neural Network Language Model*. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 5528–5531.

[16] Sepp Hochreiter and Jürgen Schmidhuber, *Long Short-term Memory*. In: Neural Computation, 9.8 (1997), pp. 1735–1780.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin, *Attention Is All You Need*, 2017, https://ar5iv.org/abs/1706.03762,

[18] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate*, 2014, Available: https://arxiv.org/abs/1409.0473,

[19] Improving Captioning for Low-Resource Languages by Cycle Consistency, 2019, https://ar5iv.org/abs/1908.07810,

[20] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, L. Zhang, *Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering*, In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 8–22 June 2018, pp. 6077–6086.

[21] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu, *BLEU: a Method for Automatic Evaluation of Machine Translation*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 2002, DOI: 10.3115/1073083.1073135, URL: https://aclanthology.org/P02-1040

[22] Ramakrishna Vedantam, C. Lawrence Zitnick, Devi Parikh, *CIDEr: Consensus-based Image Description Evaluation*, 2015, URL: https://arxiv.org/abs/1411.5726.

[23] Satanjeev Banerjee, Alon Lavie, *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, June 2005, URL: https://aclanthology.org/W05-0909.

[24] Chin-Yew Lin, *ROUGE: A Package for Automatic Evaluation of Summaries*, Text Summarization Branches Out, July 2004, Anthology ID: W04-1013, URL: https://aclanthology.org/W04-1013.

[25] Altaic languages: Turkic, Mongolic & Tungusic. Encyclopaedia Britannica. URL: https://www.britannica.com/topic/Altaic-languages.

[26] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*. International Journal of Computer Vision, 2015.

[27] ssut. *py-googletrans: Free Google Translate API for Python. Translates totally free of charge*. GitHub repository, https://github.com/ssut/py-googletrans,

[28] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. In Proceedings of the International Conference on Machine Learning, 2015. https://api.semanticscholar.org/CorpusID:1055111.

[29] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. In arXiv preprint arXiv:1409.0473, 2016. https://arxiv.org/abs/1409.0473.

[30] J. Lu, C. Xiong, D. Parikh, R. Socher, *Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning*, In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017, pp. 3242–3250.

[31] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv:1810.04805, 2018. https://arxiv.org/abs/1810.04805

[32] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *Improving Language Understanding by Generative Pre-training*, 2018. https://paperswithcode.com/paper/improving-language-understanding-by.

[33] Gemini Team Google: Rohan Anil, Sebastian Borgeaud, Yonghui Wu *Gemini: A Family of Highly Capable Multimodal Models*, 2018. https://arxiv.org/abs/2312.11805.

[34] Tuguldur, Erdene-Ochir and Gunchinish, Sharavsambuu and Bataa, Enkhbold, *BERT Pretrained Models on Mongolian Datasets*, 2019, GitHub repository, GitHub, https://github.com/tugstugi/mongolian-bert/.

[35] A. Dosovitskiy, X. Zhai, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2020, arXiv preprint arXiv:2010.11929, https://arxiv.org/abs/2010.11929.

[36] *CPTR: Full Transformer Network for Image Captioning*, 2020, arXiv preprint arXiv:2010.11929, https://ar5iv.labs.arxiv.org/html/2101.10804.

[37] Explosion AI, *spaCy: Industrial-strength Natural Language Processing*, Accessed in 2024, https://spacy.io/.

[38] Hideki Nakayama, Akihiro Tamura, Takashi Ninomiya, *A Visually-Grounded Parallel Corpus with Phrase-to-Region Linking*, 2018, https://aclanthology.org/2020.lrec-1.518.pdf

[39] Lin, Tsung-Yi and Maire, Michael and Belongie, Serge and Hays, James and Perona, Pietro and Ramanan, Deva and Dollár, Piotr and Zitnick, C. Lawrence, *Microsoft COCO: Common Objects in Context*, 2014, In Proceedings of the European Conference on Computer Vision (ECCV), https://arxiv.org/abs/1405.0312.

[40] Ehud Reiter *BLEU in Different Languages: Do not use it for German*, 2018, June 20 Observations about Natural Language Generation https://ehudreiter.com/2018/06/20/bleu-in-different-languages/.

[41] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio *Neural Machine Translation by Jointly Learning to Align and Translate* 2014, Sep 1, https://arxiv.org/abs/1409.0473

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun *Deep Residual Learning for Image Recognition* 2015, December 10, https://arxiv.org/abs/1512.03385