Universiteit Leiden

Master Computer Science

CLSEAVE: A Method for Cross-Lingual Transfer Learning in Visual Entailment

Name:	Ziyi Xu
Student ID:	s3649024
Date:	22/07/2024
Specialisation:	Data Science: Computer Science
1st supervisor:	Gijs Wijnholds
2nd supervisor:	Tessa Verhoef

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Visual Entailment is a fine-grained image-text multimodal task determining whether a text hypothesis entails an image premise. This task requires sophisticated integration of computer vision and natural language processing techniques. Contrastive learning pre-trainings have demonstrated significant success in VE, leveraging vast datasets and advanced architectures to achieve high accuracy. However, these models are primarily English-centric, highlighting a gap in cross-lingual applications due to the lack of multilingual datasets and benchmarks.

To address this, we propose the Cross-Lingual Sentence Embedding Alignment on Visual Entailment (CLSEAVE) pipeline [1]. This pipeline fine-tunes the CLIP model for VE, aligning sentence embeddings between English and translated text to create a multilingual VE model. Our experiments include languages with varying linguistic distances from English, such as German, Dutch, Japanese, Korean, and Chinese. Results indicate that CLSEAVE effectively transfers VE capabilities across languages, with performance influenced by linguistic distance and translation quality. This research advances the application of VE to a broader linguistic context, emphasizing the potential of CLSEAVE in bridging language gaps in vision-language tasks.

[1] Our code is available in Github repository.

Contents

Chapter 1

Introduction

1.1 Motivation

Visual Entailment (VE) is a novel vision-language multimodal task aimed at determining whether a text hypothesis entails an image premise in the premise and hypothesis pair (P_{image}, H_{text}), classifying their relationship into three categories: entailment, neutral, and contradiction [Xie et al., 2019]. Entailment indicates that P_{image} provides sufficient evidence to infer that H_{text} is true; contradiction means H_{text} and P_{image} conflict; neutral applies when there is insufficient evidence to make a judgment. VE is derived from Textual Entailment, also called Natural Language Inference (NLI), and requires models not only to extract features from images and text but also to integrate these two types of information for inference. The foundation for this task lies in the advances in computer vision and Natural Language Processing (NLP), where computer vision enables machines to understand images, and NLP allows machines to comprehend text. Many similar machine-learning methods have been effective in both fields. VE, a multimodal problem, extends NLI to image language inference, thus raising the requirements for models and pushing the boundaries of vision-language tasks, especially in the image-text domain.

To address VE, it's promising to get a well-performing pre-trained vision-language model and fine-tune it for the specific task. Vision-language contrastive learning is an effective pre-training objective that unifies vision and language into a shared latent space to generate universal vision-language representations [Chen et al., 2022]. CLIP [Radford et al., 2021] is a state-of-the-art contrastive learning pre-training model that bridges the gap between computer vision and NLP. Although it was designed for image

classification without predefined labels, its training on a massive dataset of image-text pairs with vast computational resources has given it a strong zero-shot capability across various vision tasks, for example, in Visual Question Answering [Shen et al., 2021] and Automatic Image Captioning [Mokady and Hertz, 2021]. Currently, the best-performing model for the VE task is OFA [Wang et al., 2022], achieving a 91.2% accuracy on the SNLI-VE test set, and prompt tuning [Yang et al., 2022b] based on OFA also surpasses 90% accuracy. These models use an encoder-decoder structure, integrating image and text inputs into a Transformer-based architecture and featuring a large number of parameters. For instance, the OFA_{large} model, which holds the highest accuracy, contains 930M parameters. While these models demonstrate excellent performance, they require significant computational resources for transfer learning. Additionally, their structure, which processes images and text together, facilitates information extraction from both types of representations within the model. Among the high-performing encoder-decoder models, CoCa [Yu et al., 2022] combines contrastive loss and captioning loss, making it highly effective. Considering these factors, we decided to attempt using CLIP directly for fine-tuning the VE task. To the best of our knowledge, such research has not been conducted before. The closest approach is using CLIP as a visual encoder in combination with vision-language pre-training for the VE task as in [Shen et al., 2021].

Currently, most research on multimodal vision and language modeling focuses on English because the most widely used multimodal datasets only include English text, and there also is a lack of multilingual evaluation benchmarks [Pfeiffer et al., 2022]. The lack of model development and application in many languages is a common trend in artificial intelligence research, raising issues of fairness and inclusivity [Bender et al., 2021]. Creating powerful models like CLIP, which is trained on 400M image-text pairs from the web, in other languages is challenging. However, CLIP's training on large and noisy web data provides it with strong generalization capabilities, and its shallow model structure, which connects visual and text encoders via a contrastive loss, gives it good alignment ability [Song et al., 2022]. Inspired by [Carlsson et al., 2022] and [Chen et al., 2023], we aimed to utilize neural machine translation to the existing English text hypotheses to avoid data scarcity issues, and achieve VE in other languages based on CLIP by cross-lingual transfer learning. Since VE's objective is to determine the relationship between premises and hypotheses, the AlignVE model [Cao et al., 2022] uses an alignment-based classifier to handle image and text features for solving VE. CLIP also adopts an alignment approach to process image and text embeddings within the same space. Thus, we applied this multimodal approach to the

multilingual problem by aligning multilingual features from different languages within a semantic space and proposing a method called Cross-lingual Sentence Embedding Alignment on Visual Entailment (CLSEAVE).

1.2 Linguistics Distance

For choosing languages for the new model, we considered their linguistic distances from English. We used the `lang2vec` database, a version of the URIEL project, which represents languages with vectors based on various linguistic distances, including six distances of typological, phylogenetic, and geographic relationships [Littell et al., 2017]. Syntactic distance involves the grammatical structure and syntax rules of languages, while featural distance quantifies language similarity by calculating the cosine similarity of feature vectors; both distances significantly impact textual transfer learning. Genetic distance is based on historical and genealogical relationships between languages and affects textual transfer learning. Phonological distance reflects the syllable structure, stress, and intonation of languages, phonetic inventory distance describes the set of phonemes in languages, and geographic distance is the physical distance between the regions where the languages are used. The latter three distances have little impact on text-based cross-lingual visual entailment. Combining language family and distance, we selected German and Dutch, both belonging to the West Germanic languages like English, and Japanese, Korean, and Chinese, which are more distant from English, to conduct the transfer learning experiments. The six types of linguistic distances between these languages and English are shown in Table 1.1.

Table 1.1: The six linguistic distances between five selected languages and English.

Distance	German	Dutch	Japanese	Chinese	Korean
Featural	0.4	0.5	0.6	0.6	0.5
Syntactic	0.42	0.49	0.57	0.57	0.62
Genetic	0.4286	0.6	1	1	1
Geographic	0.1	0	1	1	0.4
Phonological	0.3277	0.5687	0.5687	0.5687	0.4638
Inventory	0.4364	0.4861	0.5983	0.5983	0.4866

1.3 Research Questions

Our research questions can be summarized as:

1.3. Research Questions

1. How can CLIP be applied to Visual Entailment to create a robust English VE model?

2. How can we propose an effective pipeline to transfer the predictive capabilities of the English VE model to other languages? What are the performance differences between the token embedding alignment and sentence embedding alignment methods in this context?

3. How effectively is this pipeline transferring VE-solving capabilities from English to other languages with varying linguistic distances?

4. What factors influence the predictive performances of the newly transferred models, and to what extent does linguistic distance from English impact their performances?

In the following sections, we will answer the above questions step by step. Chapter 2 introduces the datasets we used, compares the pre-training models CLIP and mBERT, and reviews cross-lingual visual entailment research. Chapter 3 details our methods, describing how we built the CLIP fine-tuned Visual Entailment (CLIP-VE) model, the CLTEAVE method we tried in the first place, and how the CLSEAVE method transfers CLIP-VE to other languages. Chapter 4 documents our two types of experiments and Chapter 5 discusses the differences in experimental results. The final chapter summarizes the study and suggests future research directions.

Chapter 2

Background

In this chapter, we explore the existing research in the field of cross-lingual visual entailment. We begin by introducing the datasets developed specifically for Visual Entailment (VE). Next, we examine two powerful pre-training models: CLIP, which is a vision-language model, and mBERT, a textual model, noting their similarities and differences. We then review related studies on cross-lingual visual entailment, elaborating on the concepts of transfer learning and cross-lingual transfer learning, and discussing the application of cross-lingual transfer learning in the field of Natural Language Processing (NLP). We also highlight the crucial role of embedding alignment. Finally, we describe the methods used in our Cross-lingual Sentence Token Embedding Alignment on Visual Entailment (CLSEAVE) model and underscore the improvements it has achieved.

2.1 Data

As mentioned in our research questions, the initial step is to develop an effective English VE model. Here, we introduce the datasets required for training the model, which are associated with the tasks that VE stems from. The precursor task to VE, namely Natural Language Inference (NLI)—involves determining whether a hypothesis entails a given textual premise—a commonly used dataset is the Stanford Natural Language Inference Corpus (SNLI) [Bowman et al., 2015]. The premises in this dataset are derived from the image captions of the Flickr30k corpus [Young et al., 2014]. Flickr30k was created to study visual denotations and includes 31,783 images depicting everyday activities, events, and scenes, along with five captions per image provided by annota-

tors, amounting to a total of 158,915 captions. These captions offer relatively literal scene descriptions rather than abstract summaries or timestamps of the photographs. It has become a standard benchmark for sentence-based image descriptions. The hypotheses in the SNLI dataset were written by human workers, who were asked to provide hypotheses with entailment, neutral, and contradiction relationships for each premise, resulting in 570,152 (P_{text}, H_{text}) pairs. To further validate the dataset, 10% of the data was presented to four additional annotators, creating five labels for each pair. The gold label was determined by a majority vote among these five judgments, requiring at least three identical labels. If the agreement cannot be reached, the sentence relationship is indicated with a '−' hyphen symbol, which accounts for about 2% of the cases. The structure of SNLI leads to the SNLI-VE dataset [Xie et al., 2019] for VE tasks, which replaced the text premises in the SNLI dataset with corresponding images in Flickr30k to the captions. This allows the formation of a large number of (P_{image}, H_{text}) pairs for training and makes it the most commonly used dataset for VE tasks. The data distributions of SNLI and SNLI-VE are shown in Figure 2.1 and Table 2.1. The distribution of the development and test sets is very similar in both datasets, so only the test set is shown here.

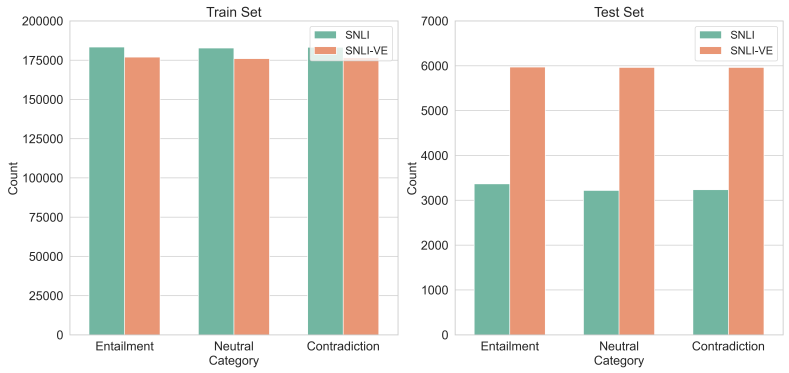

Figure 2.1: The number of entailment, neutral, and contradiction relations in the SNLI and SNLI-VE train and test sets. The SNLI train set has about 183,000 pairs in each category, while the SNLI-VE has 176,000; the SNLI test set and dev set have about 3,300 pairs in each category, while the SNLI-VE has 6,000.

It can be observed that the total number of entries in SNLI-VE is smaller than in SNLI because SNLI-VE omits data that did not reach agreement and thus lacks a gold label. Additionally, it excludes the 4k sentence pairs obtained from the Visual Genome

Table 2.1: The number of image premise and text hypothesis pairs contained in each set of the SNLI and SNLI-VE datasets.

Dataset	SNLI	SNLI-VE
Trian set	550,152	529,527
Dev set	10,000	17,858
Test set	10,000	17,901
Total	570,152	565,286

corpus [Krishna et al., 2016], which was still under construction at that time, in the SNLI training set. The SNLI-VE dataset redistributes its sets based on image premises to ensure that the development and test sets each contains 1,000 image premises and that each split set has an equal distribution of the three labels. The SNLI-VE dataset is used for fine-tuning CLIP on the VE task and validating the performance of the fine-tuned model.

2.2 Basic models

2.2.1 CLIP

CLIP, short for Contrastive Language-Image Pre-training [Radford et al., 2021], is an efficient model for addressing image-text matching problems. It uses natural language supervision to learn image representations, allowing it to avoid the need for high-quality manually labeled datasets, which are often limited in size. For example, commonly used datasets like MS-COCO [Lin et al., 2014], and Visual Genome [Krishna et al., 2016] contain around 100k images, and YFCC100M [Thomee et al., 2016]'s valid ones, and ImageNet [Deng et al., 2009] have about 15M. In contrast, CLIP was trained on a self-constructed WebImageText dataset containing 400M image-text pairs from the internet. Unlike unsupervised or self-supervised learning, CLIP not only learns image representations but also links them with language. The structure of CLIP, illustrated in Figure 2.2, is highly modular, consisting of separate image and text encoders connected only through a loss function. In contrastive learning, a batch of N image-text pairs was fed to the image and text encoders separately. CLIP was trained to predict which of the $N \times N$ possible pairings actually occurred. Only the diagonal pairs represented true matches in this $N \times N$ matrix. Therefore, the encoders were trained to maximize the cosine similarity of image and text embeddings in the multimodal embedding space for the diagonal pairs (I_i, T_i) while minimizing those for

the N^2–N off-diagonal pairs $(I_i, T_{j,i\neq j})$. In this way, the model learns which images and texts are related and which are irrelevant, thereby improving the discriminative ability [Le-Khac et al., 2020].

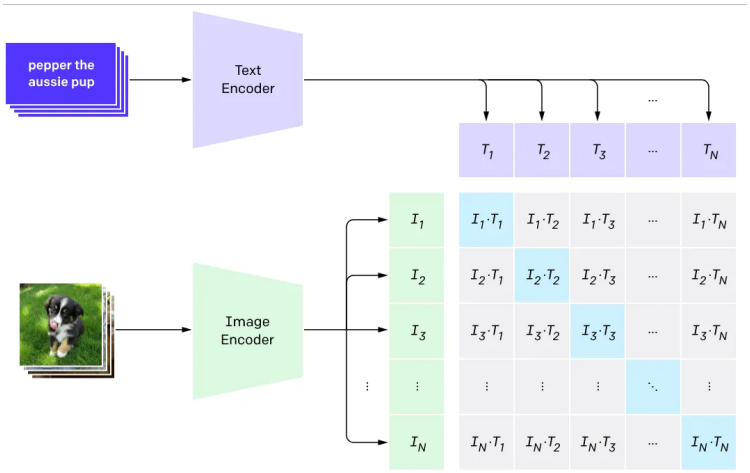

Figure 2.2: The architecture of CLIP [Radford et al., 2021]. The image and text encoder generates I and T vectors, which represent the embeddings of the image and text batch. And they form a matrix by dot product, where $I_i \cdot T_i$ represents product of the matched image and text pairs in blue. Those unmatched off-diagonal elements are in grey.

CLIP's image encoder can be either a ResNet [He et al., 2015] or a Vision Transformer (ViT) [Dosovitskiy et al., 2020]. In this study, we chose the `openai/clip-vit-base-patch16` model, with a base-sized ViT with 16×16 patches, for all used CLIP models. We chose this model because, with a similar number of parameters, the CLIP model with a ViT image encoder performs better than the one with a ResNet image encoder. In the ViT series, the `B/16` variant, which has 149 million parameters, provides greater data processing capabilities compared to the `B/32` model, which has 86 million parameters. Moreover, it does not require as much computational resources as the `L/14` model, which has 428 million parameters. It balances the trade-offs between performance and computational requirements. The CLIP text encoder is a Transformer [Vaswani et al., 2017] with GPT-2 architectural modifications [Radford et al., 2019], using a transformer encoder architecture. It is structured with 12 identical layers, each integrating an 8-head self-attention mechanism and a position-wise fully connected feed-forward network, yielding an output of 512 dimensions. The text input

to the CLIP text encoder starts with an `[SOS]` token and ends with an `[EOS]` token. The output of the topmost transformer layer represents the embeddings for each token, with the `[EOS]` token's embedding serving as the feature representation of the text sequence. This embedding is then linearly projected into the image-text multimodal embedding space, becoming the sentence embedding. Since both the CLIP image and text encoders are transformer-based, we refer to them as image and text models in the following sections to avoid confusion with the encoder module in transformers.

2.2.2 mBERT

mBERT, which stands for Multilingual Bidirectional Encoder Representations from Transformers [Devlin et al., 2019], has strong language understanding capabilities and can be used for various downstream NLP tasks, including NLI. Its structure is the same as BERT, which is a bidirectional transformer [Vaswani et al., 2017], often referred to as a 'transformer encoder'. It has 12 identical layers, each with a 12-head self-attention mechanism and a position-wise fully connected feed-forward network. The self-attention mechanism allows each position in the encoder to attend to all positions in the previous encoder layer, and its multi-head architecture captures long-range dependencies in text and allows for parallel processing. Residual connections link layers, and allow the model to selectively pass through or bypass information from the previous layer, facilitating more efficient information flow across layers. For smooth computation in residual connections, the embedding layer output dimension and hidden size are both 768. During pre-training, it utilized the entire Wikipedia dump for 104 languages, covering languages with the most expansive range of data, and applied under-sampling and over-sampling techniques to balance high-resource and low-resource languages. The input begins with an `[CLS]` token, with `[SEP]` token to separate different tokenized sequences and to end. Additionally, it conducted two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM requires the model to predict masked tokens based on their last hidden state, learning deeper token representations. NSP asks the model to determine if two text sequences are adjacent, helping the model understand sentence relationships. Specifically, we chose the `google-bert/bert-base-multilingual-cased` for this study due to its multilingual pre-trained support in German, Dutch, Japanese, Korean, and Chinese, and addresses normalization issues in languages with non-Latin alphabets.

2.2.3 Comparison

The CLIP model includes independent image and text models, making it easy to extract the text model for cross-lingual visual entailment. The CLIP text model and mBERT are both based on a bidirectional transformer, featuring two main components: the embeddings and the encoder. The CLIP text model's input comprises only the sum of token embeddings and position embeddings, without segment embeddings. Its training logic is that when people read a word and see its corresponding image, such as 'dog' and a picture of a dog, they should evoke the same concept. In the model, this is reflected in a shared embedding space where images and text describing the same object have high similarity. This principle can be applied not just to multimodal tasks but also to multilingual ones. The CLIP text model's pre-training focuses on understanding different representations without considering sentence relationships, hence it lacks segment embeddings to mark different token types. The encoders of both the CLIP text model and mBERT consist of 12 layers, and the structure of each layer is shown in Figure 2.3.

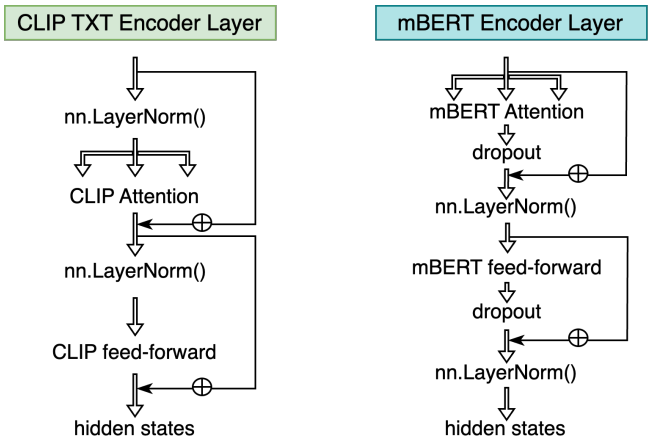

Figure 2.3: The structure of each layer of the CLIP text model encoder and mBERT encoder, and each encoder consists of two main parts: an attention mechanism and a feed-forward network. They have similar self-attention, containing Query, Key, Value and Output projections, and the same feed-forward networks with two linear functions and a GELU activation function.

Both models incorporate two blocks centering self-attention mechanism and feed-forward network but differ in connection methods. Their self-attention mechanisms build along the same logic, and feed-forward networks are exactly the same. In the

CLIP encoder layer, the process begins with normalization, then moves to the central block, and finally adds the initial state together. In mBERT, the process starts with the central block, goes through a dropout layer with the probability of 0.1, and then performs addition and normalization.

2.3 Cross-lingual Visual Entailment

Transfer learning involves extending a model thoroughly trained on a rich dataset for a specific task, and applying it to other tasks to enhance the model's generalization capabilities [Ruder et al., 2019]. Cross-lingual transfer learning is the process of adapting a model to a new language by leveraging data and tasks from a source language [Alyafeai et al., 2020]. This technique can address the lack of labeled training data for low-resource languages, thereby improving the model's ability to handle these languages and conserving computational resources. The NLP field predominantly focuses on English processing and understanding, however, enabling well-performing models to adapt to other languages can help mitigate the English-centric bias.

In NLP, cross-lingual transfer learning is widely applied. For instance, [Wang et al., 2023] attempted to use large language models for zero-shot cross-lingual summarization without fine-tuning specific language pairs. Similarly, [Abdalla and Hirst, 2017] proposed using a single linear transformation with word pairs to capture sentiment relationships between languages, facilitating cross-lingual sentiment analysis without precise translations. The XLDA [Singh et al., 2019] method, which replaces parts of the input text with its translations, was introduced to enhance model performance in cross-lingual NLI and question answering. Beyond textual information, cross-lingual transfer learning is also applied in visual-language understanding. This field is relatively new, and the limited availability of datasets has constrained research, leading most studies to focus on areas like image and video captioning and visual question answering. In cross-lingual visual grounding, [Dong et al., 2021] created a French dataset to transfer knowledge from a trained English model to a French model, achieving similar accuracy to the original model.

Embedding Alignment is a key method for cross-lingual transfer learning. By representing words or sentences from different languages in a shared semantic space, cross-lingual embeddings facilitate the transfer and sharing of knowledge and information across languages. For instance, 'dog' in English and 'hond' in Dutch are mapped to similar vectors in the semantic space. This cross-lingual transfer learning approach can be seen as optimizing similar objectives. Research indicates that model differences

in this field mainly come from the various types of embedding alignment methods used, while the choice of architecture, hyperparameters, fine-tuning, and additional techniques only produce fine-grained differences [Ruder et al., 2017]. Depending on data types, there are three embedding alignment tasks: word alignment, sentence alignment, and document alignment. Word alignment, often built with dictionaries to create parallel word corpora, is the most commonly used method. The other two methods typically require machine translation to construct parallel corpora.

In the VE field, there is limited research due to dataset constraints. In our study, we used machine translation to generate parallel corpora, which helped alleviate this problem. For cross-lingual transfer learning in VE tasks, the focus is on transferring the text hypothesis, and we aimed to select the most appropriate alignment method. The CLiCoTEA [Karoui et al., 2023] uses contextualized token alignment, extracting token embeddings of the source and target languages from fine-tuned ALBEF [Li et al., 2021] and mBERT, respectively, for teacher-student learning. Although it achieves good results in evaluation, it employs original and transferred models with consistent structures and covers a limited kind of language from IGLUE [Bugliarello et al., 2022]. The token alignment can be seen as a more granular method than word alignment. Our CLSEAVE method, on the other hand, uses the CLIP-VE model and pre-trained mBERT to extract sentence embeddings for cross-lingual visual entailment. We extracted the final hidden state projection from the special end token of a tokenized sequence, instead of token-level information. CLSEAVE successfully transfers knowledge between two structurally different models and extends cross-lingual results to languages that are less commonly used as benchmarks. We also analyzed the impact of linguistic distance and machine translation on cross-lingual visual entailment.

Chapter 3

Methodology

Our objective is to transfer the knowledge from the CLIP model, particularly its ability to handle Visual Entailment (VE) tasks, from English to other languages. By aligning the embeddings of the CLIP text model with those of textual models for different languages, we can achieve this goal, thereby saving pre-training resources and time. Additionally, we aim to investigate the impact of linguistics distance on transfer learning effectiveness. To transfer CLIP's comprehensive abilities in image and text from English to other languages, we initially planned to first transfer the pre-trained model and then fine-tune it on the VE task. However, we changed our approach due to the large size of the pre-trained model and the substantial computational resources required by multilingual-CLIP [Carlsson et al., 2022]. Instead, we fine-tuned the CLIP model on the VE task (CLIP-VE) and then transferred the resulting CLIP-VE model across languages.

Focusing on the semantic content of the text, we tried to extract semantically equivalent word pairs from sentences in different languages, and then aligned the *token embedding* of these word pairs between CLIP-VE text model and mBERT. This method is described as Method One: CLTEAVE in Section 3.2, but it performed poorly in tests. The results made us revisit and compare the two textual processing models. Based on the analysis, we extracted *sentence embedding* for alignment training. This approach yielded predictions that were comparable to or slightly better than the original model's performance. We document it in Section 3.3 Method Two: CLSEAVE.

3.1 Preprocessing & Benchmark Model

3.1.1 Data Preprocessing

We need VE datasets for transfer learning to the selected five languages. Currently, there are no human-generated multilingual training sets for VE or even for NLI tasks. XNLI [Conneau et al., 2018], which is a Cross-Lingual Natural Language Inference Evaluation Set, has released a machine-translated training set. However, it only includes $(P_{text}, H_{text}, label)$ without image ID information, making it unsuitable for matching P_{text}'s images and generating the VE dataset directly. XNLI is a crowd-sourced collection of 5,000 test pairs and 2,500 dev pairs for the MultiNLI corpus [Williams et al., 2018], extended to 15 languages by translators. Among the five languages we focus on, only German and Chinese are covered by XNLI's fifteen languages. Another cross-lingual evaluation, the IGLUE benchmark [Bugliarello et al., 2022], includes a VE validation set but only covers Arabic, Spanish, French, and Russian. Therefore, we translated the textual hypotheses from the SNLI-VE train, develop, and test sets into German, Dutch, Japanese, Korean, and Chinese using Google Neural Machine Translate [Wu et al., 2016]. This resulted in new VE datasets in five languages for transfer learning and validation to the final performances.

3.1.2 CLIP fine-tuning on Visual Entailment task

CLIP [Radford et al., 2021] is a powerful model that, through extensive training on pairs of images and texts, not only can predict the most relevant text for a given image but also possesses a significant capability in understanding both images and texts. Since CLIP focuses on the similarity between a list of images and a list of texts, it identifies the best-matching image-text pair from an image-text matrix. This structure does not directly apply to the VE problem, which is a three-class classification task. Therefore, we augmented CLIP with a three-layer MLP, resulting in the CLIP-VE model. The training structure of CLIP-VE, as shown in Figure 3.1, consists of two parts, *encoding* and *matching*. This model trained in English text premises and served as a cross-lingual transfer learning baseline.

In the encoding stage, we used the CLIP image and text models to obtain embeddings for the hypothesis and premise, respectively. The image model preprocesses images by resizing them to 224 pixels using interpolation and converting them to the RGB color format. The text model, on the other hand, can handle a maximum length of 77 tokens. Thus, there is no restriction on the input image, and the text model's

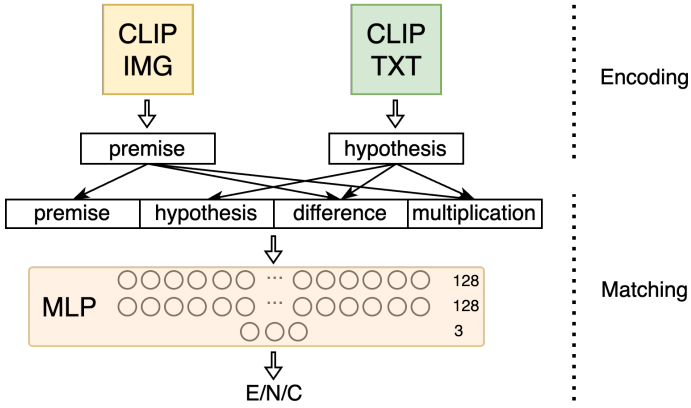

Figure 3.1: The CLIP-VE model architecture. The encoding part generates hypothesis and premise representations, while the matching part combines them as the input to a three-layer MLP, which has 128, 128, and 3 neurons in each layer respectively.

length capacity is sufficient for regular needs. The premises in the SNLI-VE dataset that we used from Section 2.1 all have lengths shorter than 77 tokens.

After obtaining the image and text features, to ensure a one-to-one correspondence between images and texts, and to move beyond the original matrix-based pairwise comparison approach, we drew inspiration from [Mou et al., 2015] and [Liu et al., 2016]. We employed the concatenation of the two representations, along with their element-wise difference and product, as the input to a three-layer fully connected multilayer perceptron. This approach helped to aggregate information and extract relationships between the premise and hypothesis. Since the output of two encoders is 512-dimensional, we can easily perform element-wise operations. The probability of the relationship $r \in \{entailment, netural, contradiction\}$ is computed by the following equations:

$$P(r|h^{hyp}, h^{pre}) = \mathbf{W}a + \mathbf{b} \tag{3.1}$$

$$a = \mathbf{MLP}(f) \tag{3.2}$$

$$f = \left[h^{hyp}, h^{pre}, |h^{hyp} - h^{pre}|, h^{hyp} \circ h^{pre}\right] \tag{3.3}$$

h^{hyp} and h^{pre} are the hypothesis and premise embedding vectors, respectively. And f is the input combination to the three-layer MLP. For other component settings, we referred to [Choi et al., 2017], applying ReLU non-linear activation to the first

two layers of perceptrons, and adding batch normalization and dropout layers both before and after them. Since PyTorch [Paszke et al., 2019] cross entropy loss function internally applies softmax and negative log-likelihood loss, it expects unnormalized logits but not the possibilities. So we passed the raw output of the third layer together with the gold label directly to compute the cross entropy loss.

During fine-tuning, we froze the image model while updating the text model and MLP. This approach balances adaptability with computational efficiency and allows the model to gain a deeper understanding of the text, making it suitable for our cross-lingual transfer learning goal. Our final CLIP-VE baseline model achieved a test loss of 0.6550 and test accuracy of 73.17%. The changes in loss and accuracy during the training and validation process are illustrated in the following Figure 3.2. We set early stopping for the training, halting when the latest validation loss was higher than the previous epoch's result. Although the validation accuracy slightly increased when the validation loss first decreased, our large training dataset makes it easy to overfit if training continues, which would degrade the model's performance. With this setting, our model trained for six epochs. For additional details on other hyperparameters, please refer to Table A.1.

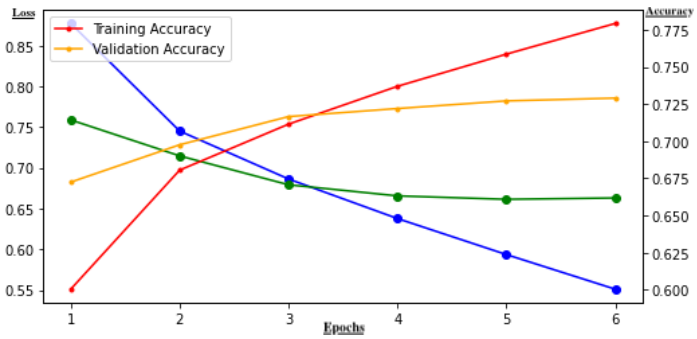

Figure 3.2: The train and validation performances of CLIP-VE. It shows the loss on the left vertical axis, and accuracy on the right vertical axis during six epochs with early stopping. The blue line is training loss, the green stands for validation loss, the red is training accuracy, and the yellow is for validation accuracy. The final training loss is 0.5511, validation loss is 0.6630, training accuracy is 77.92%, and validation accuracy is 72.89%.

3.2 Method One: CLTEAVE

CLTEAVE stands for Cross-lingual Token Embedding Alignment on Visual Entailment, which consists of three steps, with its architecture illustrated in Figure 3.3. First, we fine-tuned the CLIP model on the Visual Entailment task to obtain the CLIP-VE model as shown in Section 3.1.2. Next, we trained a multilingual autoencoder capable of converting hidden state dimensions using sentences and word pairs from five languages. Finally, we utilized token embeddings from synonymous word pairs to train the mBERT and the autoencoder. These were combined with the CLIP image model and MLP to complete the cross-lingual transfer of the CLIP-VE model.

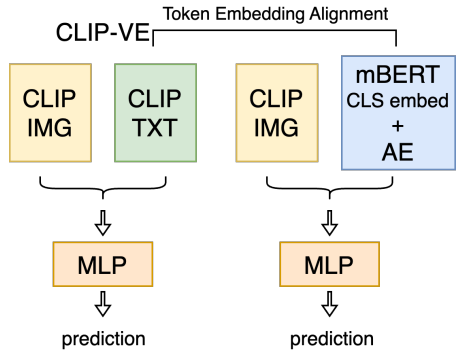

Figure 3.3: The CLTEAVE uses cross-lingual token embedding alignment to update the mBERT and the AutoEncoder together, and then combine it with the original CLIP image model and MLP to process other languages. The mBERT model with a pooling layer can extract CLS embedding, but this sentence-level information was not used in training. Instead, token-level information was used to update the new text model in blue.

3.2.1 Cross Lingual Token Embedding Alignment

Cross-Lingual Token Embedding Alignment involves two main steps: first, obtaining word alignment pairs between source and target sentences, and second, training the new model with the extracting token embeddings from these pairs.

Word Alignment

To extract token embeddings, it is essential to identify which words in two semantically equivalent sentences correspond to each other. For this purpose, we used the `awesome-align` tool [Dou and Neubig, 2021] to get word pairs. This model is designed

to extract word alignments from BERT-based models, enabling effective word alignment across different languages by leveraging BERT's powerful contextual embeddings and achieving robust performance in zero-shot settings.

The specific steps involved tokenizing the premise sentences using the model's tokenizer to obtain token IDs and a mapping between subwords and their original words. Next, the token IDs of the source and translated (target) sentences were input into the model. The similarity between these tokens was then evaluated using the dot product of their hidden states. This process resulted in a probability matrix indicating the similarity between each source subword and each target subword. From this, we selected the most similar subwords from both languages to determine subword pairs, which were then mapped back to their original words. The output was in a set of aligned word pairs, each represented as a tuple of source and target word indices.

Token Embedding

Using the five-lingual SNLI datasets generated in Section 3.1.1 and their word alignment pairs with the English SNLI datasets, we aligned the contextualized token embeddings between the CLIP-VE text model and mBERT. These token embeddings came from the last hidden state of BERT and BERT-based models, encapsulating the final feature information and providing strong text comprehension capability, usually considered as token embeddings.

The token embedding alignment was our initial approach. We came up with this idea because recent research (CCLM) demonstrated that multilingual and multimodal pretraining essentially align two different views of the same object into a common semantic space [Zeng et al., 2023]. Therefore, for our purpose, aligning synonymous token embeddings can achieve cross-lingual visual entailment. As shown in Figure 3.6, we used the mBERT with a pooling layer, in the middle of the image, as the text processor in this method. We extracted token embeddings from the processor encoder's last hidden state, which is the output of the first step in the figure. To maintain consistency between CLIP-VE and mBERT models, we did not alter the 12 hidden layers in both models. To address the mBERT output dimension that does not match that of CLIP, we added a multilingual autoencoder after mBERT's last hidden state, which will be detailed in Section 3.2.2.

In the preprocessing stage, to obtain the encoder's outputs and extract the relevant token embeddings for training specific language BERT models, we adapted the token alignment dataset and model from the CLiCoTEA pipeline [Karoui et al., 2023] to better suit our needs. We used three inputs: source language sentences, target language

sentences, and word pairs. First, we tokenized the source and target sentences separately using the CLIP and mBERT tokenizers to get token IDs and attention masks. Then, we converted word pairs into token pairs by aligning the first occurrence token in the sentences, ensuring accurate mapping. We also built unique token indexes for each batch to track the token positions in sentences, helping to maintain the correct alignment between source and target tokens throughout training. The yellow triangle in Figure 3.4 depicts these processes.

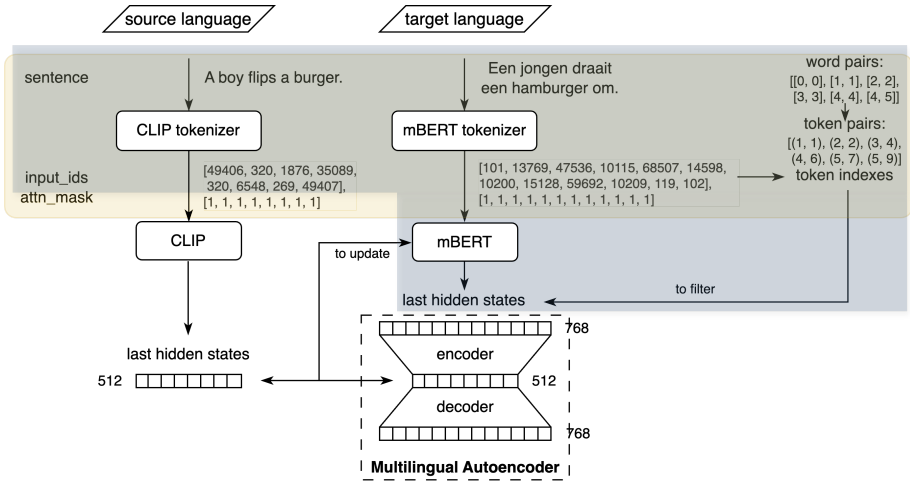

Figure 3.4: The architecture for the cross-lingual token embedding alignment and Multilingual Autoencoder. The yellow rectangular box is the data loading and preprocessing of the former, and the blue shadow shows the latter.

During the training phase, the source language token IDs and attention masks were fed into the fine-tuned CLIP-VE text model, while the target language equivalents were provided to the mBERT model. Aligned token embeddings were extracted from the last hidden states of both models using token indexes from the preprocessing step. We froze the CLIP text model, and utilized the MSE loss between the token embeddings of both models to update mBERT and the multilingual autoencoder. To streamline our training process, we used PyTorch-Ignite [Fomin et al., 2020], a high-level library that simplifies and enhances training workflows in PyTorch. Ignite's handlers automatically saved the two most efficient models, while the Engine and Events classes allowed us to define training loops and attach custom behaviors like logging and learning rate adjustments. Additionally, the ProgressBar helped us track training progress visually, and integration with Weights & Biases [Biewald, 2020] enabled seamless logging and

visualization of training metrics. This approach reduced boilerplate code and improved the efficiency and maintainability of our training process.

3.2.2 Multilingual Autoencoder

The final and simplest module we built in CLTEAVE was the multilingual autoencoder. This unsupervised model encoded 768-dimensional input into 512 dimensions and then decoded it back to the original size. We needed this because the token embeddings of the CLIP-VE text model were 512 dimensions, while those of mBERT were 768. Using the encoder from the multilingual autoencoder, we converted mBERT's last hidden states to 512 dimensions, matching the size of CLIP-VE's hidden states.

The architecture of the multilingual autoencoder was straightforward. Its encoder and decoder each consisted of a symmetrical single-layer perceptron. This perceptron included only a linear layer and a ReLU activation function. We chose a linear layer instead of a convolutional one or other methods because the target dimensionality was greater than half of the input dimension and much smaller than it. Thus, padding and pooling were not suitable for dimensionality reduction in this context.

The autoencoder dataset was built based on the token embedding dataset in the last subsection. We still needed to input source sentences, target sentences, and word pairs to generate input IDs and attention masks for the target language, and token pairs. In addition to the original token embedding dataset, we loaded the pre-trained mBERT model to process the target language information and get the last hidden state. This data preprocessing step is depicted by the blue shadow in Figure 3.4.

In the cross-lingual transfer learning setup, the autoencoder acted as a data preprocessor, taking the 768-dimensional last hidden states as input and output its 512-dimensional projection. Its specific place in the method is framed by dotted lines in Figure 3.4. We input combined corpora from five languages to our autoencoder, allowing it to handle multiple languages simultaneously. And we used the Mean Squared Error (MSE) between its input and output to update. The training loss of this multilingual autoencoder was 0.1529.

3.3 Method Two: CLSEAVE

CLSEAVE, short for Cross-lingual Sentence Embedding Alignment on Visual Entailment, has two steps, and the architecture of this method is illustrated in Figure 3.5. First, we employed the fine-tuned CLIP-VE model, as detailed in Section 3.1.2. Then,

following the structure of the CLIP text model, we constructed a student model that used mBERT to extract text information and trained with cross-lingual sentence embedding alignment. Finally, replacing the original text model with this new student model enabled it to process premises in other languages.

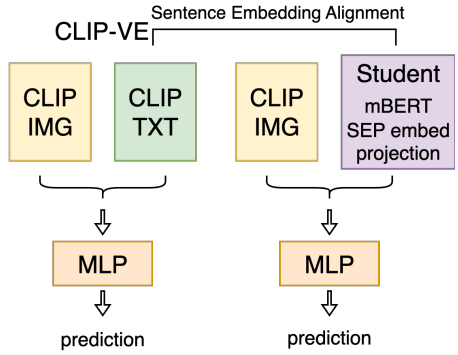

Figure 3.5: The CLSEAVE uses cross-lingual sentence embedding alignment to train the student model to process other languages, then combine it with the original CLIP image model and MLP. The student model in purple consists of the mBERT model, a SEP sentence embedding extraction, and a linear projection.

3.3.1 Cross Lingual Sentence Embedding Alignment

Upon obtaining the CLIP-VE model in Section 3.1.2, we selected the mBERT model for cross-lingual visual entailment, aiming to extend the task-solving capabilities of the CLIP-VE model to other languages. We chose mBERT because the CLIP text model is BERT-based, so they share similar characteristics. Moreover, mBERT is proficient in understanding multiple languages. After the embeddings and encoder modules, both models generate a last hidden state, which provides a comprehensive depiction of the input text and contains contextualized representations for each token that can serve as token embeddings. They also include sentence-level information: in mBERT, the `[CLS]` token embedding in the last hidden state summarizes the entire input sentence, making it commonly used for classification tasks. In CLIP, the `[EOS]` token embedding in the last hidden state is extracted, allowing the model to process and consider all preceding tokens.

Our experiments showed that the student model, constructed by following the CLIP processing steps on mBERT, achieved the best performance. Specifically, we used the

3.3. Method Two: CLSEAVE

mBERT [SEP] token's last hidden state projection as the sentence embeddings and aligned it with the projection from the CLIP [EOS] token's last hidden state for transfer learning. In our model, there was only one sentence in input, so the only [SEP] token in mBERT acts as a special terminal token. This also partially explains the ineffectiveness of the [CLS] embedding, since the classic usage of [CLS] is like in NLI tasks, BERT connects two sentences using [SEP] token and uses [CLS] token's final hidden state for classification. VE task attempts to infer the relationship between (P_{image}, H_{text}), so there is only one sentence of text input. Our model implies aligning the terminal token embeddings, which contain comprehensive sentence information, can effectively facilitate cross-lingual visual entailment.

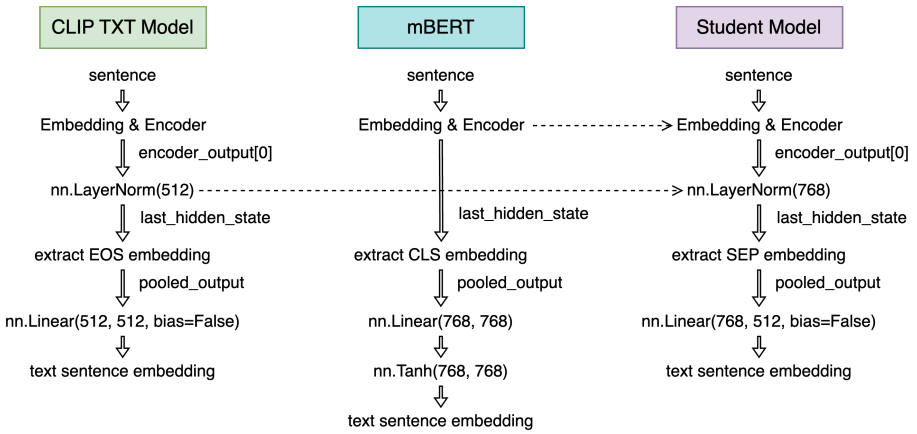

Figure 3.6: Comparison of the architecture between the CLIP text model, mBERT with a pooling layer, and our assembled student model.

The structure of the student model is shown in Figure 3.6. We used BERT as the text processor but did not use its pooling layer and activation. Instead, we followed the CLIP text model's procedures to obtain pooled output and text embeddings. The specific steps are: first, we added layer normalization to the last hidden state of mBERT and then extracted the [SEP] embedding instead of the [CLS] embedding. A crucial step here involved reducing the dimensionality of the mBERT outputs, which are 768 dimensions, to match the 512-dimensional projection in CLIP. This was achieved by applying a linear layer for dimension reduction, instead of configuring a linear layer of the same dimension as both the original CLIP and mBERT models and then adding an autoencoder as done in the CLTEAVE method. Thus, we obtained the sentence embedding.

We paired sentences from the translated dataset in Section 3.1.1 with synonymous sentences from the original SNLI-VE dataset in Section 2.1 as input and froze the CLIP-VE text model. The entire student model was updated using the MSE loss between the sentence embeddings from the CLIP text model and the student model. After obtaining the text processing model for other languages, replacing the English CLIP-VE text model with it achieved favorable VE test performances, which will be further elaborated in Section 4 Experiments.

Chapter 4

Experiments & Results

4.1 Sentence Embedding vs. Token Embedding

Our CLSEAVE method extracts sentence embedding from English sentences and their parallel corpora, transferring the CLIP text model's understanding of English to other languages. This constructs a new student model that utilizes mBERT as the text processor and projects the last hidden state of the sequence's final token into an embedding space. We then combine it with the CLIP image model and MLP classification to achieve cross-lingual visual entailment. In this experiment, we replaced the text processor in CLIP-VE with the mBERT trained with the CLTEAVE method described in Section 3.2. The module combined the token embedding alignment-trained mBERT with the CLIP-VE text model's layer normalization and linear projection. Then, we compared its performance with the model trained by CLSEAVE.

Since BERT's output is 768 dimensions and CLIP's is 512, we padded the 512-dimensional weights and bias parameters to 768 dimensions. We also attempted to use the decoder part of the Autoencoder from Section 3.2.2 for dimensional transformation.

We conducted this experiment in German, and the results show that using both dimensional conversion methods, the test prediction accuracy for the combined module is 33.32%. Given that VE has three classification categories, this result is the same as a random baseline. The results obtained using this approach were identical to those achieved by directly using the CLTEAVE method, which involves the combination of mBERT and an autoencoder. Since this method did not lead to any improvement, we decided not to pursue further experiments with it and instead continue with the CLSEAVE method. The results for the CLSEAVE approach are presented in the next

section.

4.2 CLSEAVE to Five Languages

In this experiment, we input the SNLI-VE and SNLI-VE$_{other\ language}$ datasets and used the CLSEAVE pipeline to perform transfer learning on CLIP-VE and create CLIP-VE$_{other\ language}$ models. We selected five languages for this purpose: Dutch and German, which are linguistically close to English, and Japanese, Korean, and Chinese, which are more distant.

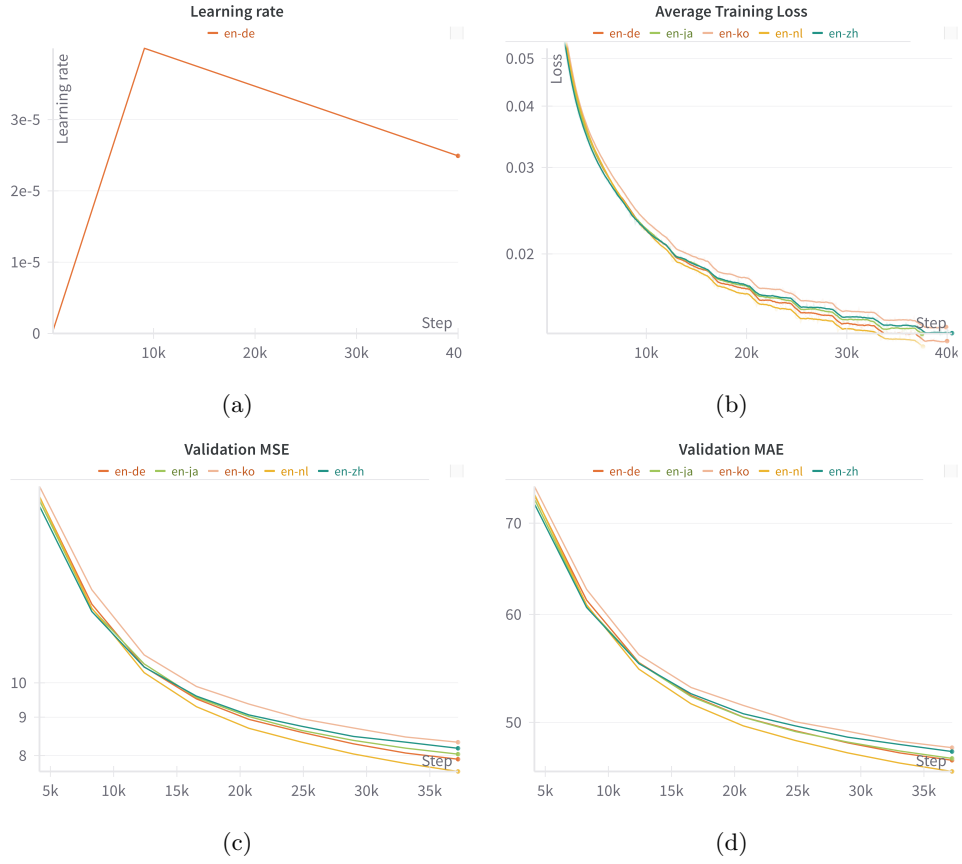

Figure 4.1: Figure (a) shows the changes in the learning rate to the German model, Figure (b) shows the average training MSE to five language models, Figure (c) and (d) show the validation MSE and MAE during nine epochs.

We conducted parameter experiments for epochs, batch size, learning rate and

scheduler, gradient accumulation, and gradient clipping. We chose the optimal parameters for all subsequent experiments, which set epochs=22 with early stops at 9 epochs, batch size=128, learning rate=1×10^{-4}, a linear scheduler with warmup, and without gradient accumulation or gradient clipping. The experimental results are shown in the above figure. For ease of comparison, the learning rate changes in all five language experiments were consistent. CLIP-VE$_{German}$ is used as an example here. We performed logarithmic scaling on the y-axis of the average training loss and ignored outliers in the chart scaling for clarity. At the end of the training, the training loss for each language was between 0.013 and 0.014. Due to the logarithmization, the minimum y-axis value is greater than 0, and the long tail of the curve extends below the y-axis. We also applied log scaling to the y-axis for validation MSE and MAE to better display relative relationships between language models. The MSE for each language model was between 7.6 and 8.3, and the MAE was between 46 and 48, with the alignment training target being MSE.

Table 4.1: Test loss and test accuracy for original English CLIP and the five language transfer models

Language	Test loss	Test accuracy
English	0.6550	73.17%
German	0.6761	73.19%
Dutch	0.6853	71.35%
Japan	0.7149	70.82%
Chinese	0.7161	69.94%
Korean	0.7391	68.75%

As shown in Figure 4.1, the final transfer learning results on the validation set for each language model are quite similar, from best to worst are Dutch, German, Japanese, Chinese, and Korean. CLIP-VE$_{Dutch}$ had an MSE of 7.619 after 9 epochs, while CLIP-VE$_{Korean}$ had 8.335. The training MSE of each model was around 0.014, but the validation set was around 8, showing a significant difference. We also observed that when the validation MSE decreased during training, the test accuracy did not necessarily increase, indicating that the data is very complicated, and the transferred model is prone to overfitting on certain portions of the corpus. According to Table 4.1, the test set performance ranking is German, Dutch, Japanese, Chinese, and Korean. Unlike the validation set results, CLIP-VE$_{German}$ became the best-performing model, with accuracy slightly exceeding that of the English source model, while CLIP-VE$_{Dutch}$ dropped from first to second.

Chapter 5

Discussion

This chapter will discuss the experimental results and answer research questions individually.

For question 1, we combined the representations obtained from the CLIP image and text models element-wise and fed them into a three-layer classification MLP, creating the CLIP-VE model. The specific model structure is detailed in Section 3.1.2. The CLIP-VE model achieved a test accuracy of 73.17%, surpassing the EVE-Image model (71.56%) [Xie et al., 2019] and the AlignAE model (72.45%) [Cao et al., 2022] designed specially for the VE task.

For question 2, we constructed a student model by combining the extracted terminal token's hidden state method and the linear projection from the CLIP text model with mBERT, using CLSEAVE to train the CLIP-VE$_{other\ language}$ models. The specific steps of the CLSEAVE pipeline are outlined in Section 3.3. From Experiment 1, it is evident that the CLTEAVE method cannot transfer the predictive capabilities of the CLIP-VE model. The test accuracy of the CLTEAVE transferred German model is 33.32%, while that for CLIP-VE is 73.17%, for CLIP-VE$_{German}$ model trained by CLSEAVE is 73.19%, and the average performance of CLIP-VE$_{other\ language}$ models is 70.81%. Despite using the parameters of the source model, aligning the token embeddings from the final layer outputs of the CLIP text model and mBERT did not provide sufficient information for transfer learning. This indicates that token embedding alignment is unsuitable for cross-lingual visual entailment with CLIP, and sentence embeddings are ideal for this task.

For question 3, according to the results in Figure 4.1, the German and Dutch models perform better than the Japanese, Chinese, and Korean models. The CLIP-

VE$_{German}$ even slightly surpasses the CLIP-VE model, achieving a test accuracy of 73.17%, and the CLIP-VE$_{Dutch}$'s accuracy is only 1.82% lower than the source model. Languages that are less closely related to English and have greater linguistic distances show poorer performance after transfer. The CLIP-VE$_{Janpanese}$ and CLIP-VE$_{Chinese}$ models both have accuracies around 70%. The CLIP-VE$_{Korean}$ model performs the worst in our experiments, with a test accuracy of 68.75%.

For question 4, each language's cross-lingual visual entailment learning effectiveness is influenced by the performance of the CLIP-VE model, hyperparameter settings, neural machine translation, and linguistic distance. Since we used the same CLIP-VE source model and hyperparameters in the experiments, the differences in results are primarily due to machine translation quality and linguistic distance from English. As the training and evaluation datasets were machine-translated, and no suitable Visual-Language datasets are available, we cannot rule out the impact of machine translation quality on the performance of CLIP-VE$_{other\ languages}$ models.

The quality of machine translation is affected by the richness of bilingual corpora. The five languages we selected are all considered high-resource languages. Among them, German, Japanese, and Chinese are classified in the most resource-rich category, 5, according to the number of language resources by [Joshi et al., 2020]. These languages have a dominant online presence and receive massive industrial and governmental investments in developing resources and technologies, making them rich-source languages. Dutch and Korean are classified in the fourth category; they have abundant linguistic resources, but comparatively limited labeled data. Among the 2485 classified languages, only seven are in the most resource-rich, 5 category, while 18 belong to the very resource-rich, 4 category.

As shown in Table 1.1, in terms of linguistic distance, German and Dutch are the first and second most similar languages to English, aligning with our experimental results. Japanese and Chinese have the same featural, syntactic, and genetic distances from English in the URIEL dataset [Littell et al., 2017]. The cross-lingual transfer learning results for these two languages are also very similar, with Japanese performing slightly better than Chinese, ranking third and fourth respectively. Korean, in comparison to Japanese and Chinese, is syntactically more distant from English but is closer in featural distance. The combined impact of linguistic distance and machine translation quality shows that Korean has the poorest cross-lingual visual entailment results among the five languages. This might be because large-scale machine translation parallel databases for Korean and English are still under development, whereas databases for Chinese [Tian et al., 2014] and Japanese [Morishita et al., 2022] have

already been publicly released.

Overall, our transfer experiments achieved good accuracy results, with differences within 5% of the source model. As the selected languages are high-resource, the model performance differences are mainly influenced by linguistic distance.

Chapter 6

Conclusion

6.1 Summary

This thesis proposes the Cross-Lingual Sentence Embedding Alignment on Visual Entailment (CLSEAVE) pipeline, offering an efficient cross-lingual transfer learning framework. To utilize this framework with CLIP, we first equipped it for VE tasks by adding a classification MLP after its separate image and text models and fine-tuning it into the CLIP-VE model. We aligned the sentence embeddings extracted from parallel corpora by the CLIP text model and the student model, enabling the student model, centered on mBERT for text processing, to gain equivalent processing capabilities and thus achieve cross-lingual transfer learning. In this process, we consider the hidden state of the final token in the tokenized sequence as a comprehensive feature of the sequence and project it to be the sentence embedding. Apart from CLSEAVE, we also explored CLTEAVE, which uses the last hidden state as token embeddings for transfer learning.

Since there is currently no available multilingual textual training set or multilingual vision-language test set, we constructed SNLI-VE datasets in German, Dutch, Japanese, Korean, and Chinese using neural machine translation for our experiments. The results demonstrate that CLSEAVE can transfer the CLIP-VE model to other languages with similar zero-shot performance, and the model's performance in different languages is mainly influenced by linguistic distance from the source language and also by the quality of the machine translation.

Due to the lack of multimodal multilingual datasets, research on cross-lingual visual entailment is limited, and languages are restricted to those in existing benchmark test

sets. Our experiments show that CLSEAVE can achieve results comparable to the source model on machine-translated datasets. We included German and Dutch, which belong to the same West Germanic language group as English, as well as Japanese, Korean, and Chinese, which have a greater linguistic distance from English. This not only broadens the scope of target languages for cross-lingual visual entailment but also indirectly indicates that CLSEAVE can adapt to various languages. CLSEAVE does not rely on identical model structures; we used it to transfer VE capabilities from CLIP to mBERT, which has a different structure. Additionally, it only takes two and a half hours to complete a training on a single GPU, saving a significant amount of time compared to training a pre-trained model for a new language.

6.2 Future Work

We initially planned to apply CLSEAVE for cross-comparison, transferring not only from English CLIP-VE to German, Dutch, Japanese, and Korean, as shown in this thesis, but also from a fine-tuned Chinese-CLIP [Yang et al., 2022a] to these four languages. This would allow us to compare the effects of transferring from English and Chinese source models to languages that are either closer to or farther from their respective linguistic distances. However, due to limited time, we could not train the new fine-tuned Chinese-CLIP model or perform the transfer training for each language. Nevertheless, we completed preliminary scripts for training the Chinese-CLIP-VE model, which can serve as a foundation for future work.

Furthermore, the training and evaluation in this thesis were conducted on translated datasets. Using a test set translated or corrected by a human translator could allow us to compare machine translation evaluation with more accurate evaluation, providing insights into the impact of machine translation on the model. This would help determine whether better human-generated or translated training sets can improve the model. We can also use image generation techniques to supplement textual datasets into visual-textual datasets. Additionally, we hope to apply CLSEAVE to low-resource languages and other Vision-Language Pre-training models, which would both validate the method's practicality and help optimize it.

Appendix A

Table A.1: The hyperparameters used for models in the thesis.

Model	CLIP-VE	CLSEAVE	AutoEncoder
Learning rate	1×10^{-6}	4×10^{-5}	1×10^{-5}
weight decay	0	0	1×10^{-6}
Batch	64	128	256
Epoch	6	9	20
Loss function	CrossEntropy	MSE	MSE
Optimizer	Adam	AdamW	Adam

Table A.2: The MSE changes during nine training epochs in five language-transferred models.

Epochs	1	2	3	4	5	6	7	8	9
Dutch	17.72	12.58	10.32	9.29	8.70	8.34	8.04	7.81	7.62
German	17.70	12.73	10.50	9.52	8.94	8.59	8.29	8.06	7.91
Janpanese	17.50	12.56	10.60	9.56	9.02	8.64	8.38	8.18	8.04
Chinese	17.20	12.45	10.50	9.60	9.06	8.76	8.48	8.34	8.18
Korean	18.28	13.31	10.90	9.89	9.37	8.96	8.71	8.47	8.33

Table A.3: The MAE changes during nine training epochs in five language-transferred models.

Epochs	1	2	3	4	5	6	7	8	9
Dutch	73.53	60.99	54.71	51.60	49.71	48.49	47.49	46.68	46.03
German	73.41	61.47	55.32	52.33	50.45	49.30	48.30	47.49	46.90
Janpanese	72.93	60.84	55.28	52.23	50.44	49.23	48.37	47.64	47.05
Chinese	72.34	60.72	55.22	52.48	50.74	49.71	48.77	48.17	47.60
Korean	74.50	62.60	56.06	53.04	51.46	50.05	49.26	48.42	47.91

Bibliography

Mohamed Abdalla and Graeme Hirst. Cross-lingual sentiment analysis without (good) translation. In Greg Kondrak and Taro Watanabe, editors, *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 506–515, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-1051.

Zaid Alyafeai, Maged S. Alshaibani, and Irfan Ahmad. A survey on transfer learning in natural language processing. *ArXiv*, abs/2007.04239, 2020. URL https://api.semanticscholar.org/CorpusID:220404708.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, E. Ponti, and Ivan Vulic. Iglue: A benchmark for transfer learning across modalities, tasks, and languages. *ArXiv*, abs/2201.11732, 2022. URL https://api.semanticscholar.org/CorpusID:246294502.

Biwei Cao, Jiuxin Cao, Jie Gui, Jiayun Shen, Bo Liu, Lei He, Yuan Yan Tang, and James Tin-Yau Kwok. Alignve: Visual entailment recognition based on alignment relations. *IEEE Transactions on Multimedia*, 25:7378–7387, 2022. URL https://api.semanticscholar.org/CorpusID:253546015.

Fredrik Carlsson, Philipp Eisen, Faton Rekathati, and Magnus Sahlgren. Cross-lingual and multilingual CLIP. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6848–6854, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.739.

Feilong Chen, Duzhen Zhang, Minglun Han, Xiuyi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20:38–56, 2022. URL https://api.semanticscholar.org/CorpusID:246996617.

Guanhua Chen, Lu Hou, Yun Chen, Wenliang Dai, Lifeng Shang, Xin Jiang, Qun Liu, Jia Pan, and Wenping Wang. mCLIP: Multilingual CLIP via cross-lingual transfer. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13028–13043, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.728. URL https://aclanthology.org/2023.acl-long.728.

Jihun Choi, Kang Min Yoo, and Sang goo Lee. Learning to compose task-specific tree structures. In *AAAI Conference on Artificial Intelligence*, 2017. URL https://api.semanticscholar.org/CorpusID:24462966.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. URL https://aclanthology.org/D18-1269.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. URL https://api.semanticscholar.org/CorpusID:57246310.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:52967399.

Wenjian Dong, Mayu Otani, Noa Garcia, Yuta Nakashima, and Chenhui Chu. Cross-lingual visual grounding. *IEEE Access*, 9:349–358, 2021. doi: 10.1109/ACCESS.2020.3046719.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. URL https://api.semanticscholar.org/CorpusID:225039882.

Zi-Yi Dou and Graham Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.181. URL https://aclanthology.org/2021.eacl-main.181.

V. Fomin, J. Anmol, S. Desroziers, J. Kriss, and A. Tejani. High-level library to help with training neural networks in pytorch. https://github.com/pytorch/ignite, 2020.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. URL https://api.semanticscholar.org/CorpusID:206594692.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560.

Yasmine Karoui, Rémi Lebret, Negar Foroutan, and Karl Aberer. Stop pre-training: Adapt visual-language models to unseen languages. *ArXiv*, abs/2306.16774, 2023. URL https://api.semanticscholar.org/CorpusID:259287284.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123: 32 – 73, 2016. URL https://api.semanticscholar.org/CorpusID:4492210.

Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020. doi: 10.1109/ACCESS.2020.3031549.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:236034189.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. URL https://api.semanticscholar.org/CorpusID:14113767.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2002.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *ArXiv*, abs/1605.09090, 2016. URL https://api.semanticscholar.org/CorpusID:12305768.

Ron Mokady and Amir Hertz. Clipcap: Clip prefix for image captioning. *ArXiv*, abs/2111.09734, 2021. URL https://api.semanticscholar.org/CorpusID:244346239.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6704–6710, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.721.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Recognizing entailment and contradiction by tree-based convolution. *ArXiv*, abs/1512.08422, 2015. URL https://api.semanticscholar.org/CorpusID:1313283.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703, 2019. URL https://api.semanticscholar.org/CorpusID:202786778.

Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. xGQA: Cross-lingual visual question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2497–2511, Dublin, Ireland, May 2022. Association for Computational Linguistics.

doi: 10.18653/v1/2022.findings-acl.196. URL https://aclanthology.org/2022.findings-acl.196.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL https://arxiv.org/abs/2103.00020.

Sebastian Ruder, Ivan Vulic, and Anders Søgaard. A survey of cross-lingual word embedding models. *J. Artif. Intell. Res.*, 65:569–631, 2017. URL https://api.semanticscholar.org/CorpusID:26127787.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In Anoop Sarkar and Michael Strube, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-5004. URL https://aclanthology.org/N19-5004.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *ArXiv*, abs/2107.06383, 2021. URL https://api.semanticscholar.org/CorpusID:235829401.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. XLDA: cross-lingual data augmentation for natural language inference and question answering. *CoRR*, abs/1905.11471, 2019.

Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. CLIP models are few-shot learners: Empirical studies on VQA and visual entailment. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6088–6100, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.421. URL https://aclanthology.org/2022.acl-long.421.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, jan 2016. ISSN 0001-0782. doi: 10.1145/2812802. URL https://doi.org/10.1145/2812802.

Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. UM-corpus: A large English-Chinese

parallel corpus for statistical machine translation. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1837–1842, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/774_Paper.pdf.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:13756489.

Jiaan Wang, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. Zero-shot cross-lingual summarization via large language models. In Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini, editors, *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 12–23, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.newsum-1.2. URL https://aclanthology.org/2023.newsum-1.2.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:246634906.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016. URL https://api.semanticscholar.org/CorpusID:3603249.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

An Yang, Junshu Pan, Junyang Lin, Rui Men, Yichang Zhang, Jingren Zhou, and Chang Zhou. Chinese clip: Contrastive vision-language pretraining in chinese. *ArXiv*, abs/2211.01335, 2022a. URL https://api.semanticscholar.org/CorpusID:253254967.

Han Yang, Junyang Lin, An Yang, Peng Wang, Chang Zhou, and Hongxia Yang. Prompt tuning for generative multimodal pretrained models. *ArXiv*, abs/2208.02532, 2022b. URL https://api.semanticscholar.org/CorpusID:251320445.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://api.semanticscholar.org/CorpusID:248512473.

Yan Zeng, Wangchunshu Zhou, Ao Luo, Ziming Cheng, and Xinsong Zhang. Cross-view language modeling: Towards unified cross-lingual cross-modal pre-training. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5731–5746, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.315. URL https://aclanthology.org/2023.acl-long.315.