University of Science and Technology of Hanoi

Information and Communication Technology Department

Bachelor Thesis

Academic year: 2018 - 2021

Hateful memes classification

Author:

VU Dinh Anh

USTHBI9-037

Supervisor:

Dr. TRAN Hoang Tung, USTH ICT Lab

Hanoi - July 26, 2021

Attestation

I, VU Dinh Anh, hereby attest that my thesis doesn't contain plagiarism (copy/paste) from other sources without citation nor referenced.

If plagiarism actions are caught, I perceive the consequences that my thesis and results will not be accepted. In that case, I accept any penalty from the board and the university.

July 26, 2021
Signature

Vũ Đinh Anh

Acknowledgments

Firstly, I would like to value and respect my research supervisor Dr. TRAN Hoang Tung for accepting me and my topic as well as guiding during the internship. Secondly, my thanks goes Dr. TRAN Giang Son for granting permission to utilize USTH ICT's servers. Thirdly, I thank researchers from Microsoft, Facebook, Google and other organizations for their publication in term of source code and corresponding materials. Moreover, I thank members of Stackoverflow, Github and other forums for being supportive. Last but not least, I express my great respect and love to my parents who raise me up and support me throughout my life. Sincerely thank!

Abstract

On the internet, hateful multimodal information is being spread rapidly by people. One of an internet culture form is meme which contains both images and texts. Those memes can affect negatively people. Therefore, multimodal machine learning models are in demand to filter those memes out. In this work, top models in Facebook Hateful Memes challenge are referred to know their ideas. With that influence, multiple directional attention was created to allow UNITER to understand many data channels simultaneously. Moreover, many experiments related to image caption, paraphrased text and context were carried out. The best model got 0.8026 AUC ROC and 0.7510 Accuracy which is above 5th place in the challenge. At the end, 6 ideas for experiments in future are brainstormed.

Keywords: meme, computer vision, natural language processing, multi-modal machine learning, classification.

Contents

Attestation i

Acknowledgments iii

Abstract v

Contents vii

Acronyms ix

List of Tables xi

List of Figures xiii

1 Introduction 1
1.1 Context, motivation and objectives 1
1.2 Thesis structure 2

2 Hateful Memes Challenge 3
2.1 Dataset 3
2.2 Performance metrics 4
2.2.1 Accuracy 4
2.2.2 Area Under the Curve of the Receiver Operating Characteristic . . 5
2.3 Winning models 6

3 Methodology 9
3.1 Image feature extraction 9
3.2 Image captioning 10
3.3 Text paraphrasing 11
3.4 Image context addition 11
3.5 UNITER with multiple directional attention 12

4 Experiments 15

5 Results 19
5.1 Internal comparison 19
5.2 External comparison 21

6 Conclusion and future work 23
6.1 Conclusion 23
6.2 Future work 23

Bibliography 25

Appendices 29
.1 Figures 31

Acronyms

AI Artificial Intelligence. 1

AUC Area Under the Curve. 6

AUC ROC Area Under the Curve of the Receiver Operating Characteristic. 1, 5, 17, 19, 21, 22

EF Early Fusion. 6

FPR False Positive Rate. 6

LF Late Fusion. 6

MDA Multiple Directional Attention. 12, 15, 16, 20, 22, 23

MMML MultiModal Machine Learning. 1, 2, 6, 23

ROC Receiver Operating Characteristic. 6

TPR True Positive Rate. 6

List of Tables

2.1 Confusion matrix [3] 5
2.2 Type of prediction with the threshold 5

5.1 Experiment phase 1 result 19
5.2 Experiment phase 2 result 20
5.3 Experiment phase 3 result 20
5.4 Experiment phase 4 result 21
5.5 Comparison between the baseline and the best model 21
5.6 With other papers' models in the challenge 22

List of Figures

1.1 Example of memes [2] 2

2.1 A hateful meme - File name: `10746.png` 4
2.2 A non-hateful meme - File name: `08659.png` 4
2.3 Distribution of labels with corresponding probabilities with alterable threshold 5
2.4 Receiver Operating Characteristic curve 6

3.1 The overview of methodologies 9
3.2 Bounding boxes of `10746.png` 10
3.3 Bounding boxes of `08659.png` 10
3.4 ImgCap's architecture 11
3.5 The architecture of UNITER and MDA with 3 data channels 13
3.6 Triple cyclic attention 13
3.7 Quadruple attention 13
3.8 Hextuple attention 13

6.1 A grayscale meme - File name: `53102.png` 24
6.2 Colorized Figure 6.1 24
.1.1 The overview of methodologies 32

Chapter 1

Introduction

1.1 Context, motivation and objectives

Nowadays, due to the availability of internet and the presence of many social media platforms, more and more people spend much time online. As of writing, there are 4.72 billion people accessing internet [1]. People share different pieces of information in divergent domains. The content's tones can be useful, wholesome, misleading, hateful, etc. Hateful content is affecting people's mental health and world-view negatively. In cyberculture, the format of content, which is spread fast and vastly, is meme. A meme is a way to entertain people with visualization by using a still or animated image [1]. Moreover, a meme is often made of a single image and additional embedded texts. Recently, hateful memes have become a major concern for people. Therefore, it is needed to have an automatic system to filter large amount of diverse memes.

Modality often defines as the way for something to express or to be perceived. For example, ”the dog is barking” can be displayed by an image/a scene or written as text or said by someone/something. That example can appear in 3 modalities which are vision, non-verbal linguistics and audio. MultiModal Machine Learning (MMML) targets creating models that are able to learn from and correlate with different modalities. MMML is first few steps to make Artificial Intelligence (AI) applications understand our world. In this work's scope, MMML models, which can relate visual and linguistic modalities, are centred. The models could be known as image-text model or visual-linguistic model or V+L model.

Facebook AI opened Hateful memes challenge in 2020 and fully publicised dataset in 2021 [2]. The competition consists of Phase 1 - research stage and Phase 2 - compete stage. Phase 2 **determined** final ranking. It's aimed to distinguish non-hateful and hateful memes. Therefore, this is a binary classification problem. Two performance metrics are Area Under the Curve of the Receiver Operating Characteristic (AUC ROC); accuracy. It's noted that AUC ROC is the primary criteria, while Accuracy is the secondary one. All following metrics' values are decimal from 0.0 to 1.0. The goal is to reach high as much as possible.

This work's objectives are summarized in as follows:

[1] https://datareportal.com/global-digital-overview

[2] https://hatefulmemeschallenge.com/

- To study MultiModal Machine Learning.
- To get involved in Facebook Hateful Memes challenge.
- To propose improvement(s) to MMML models.

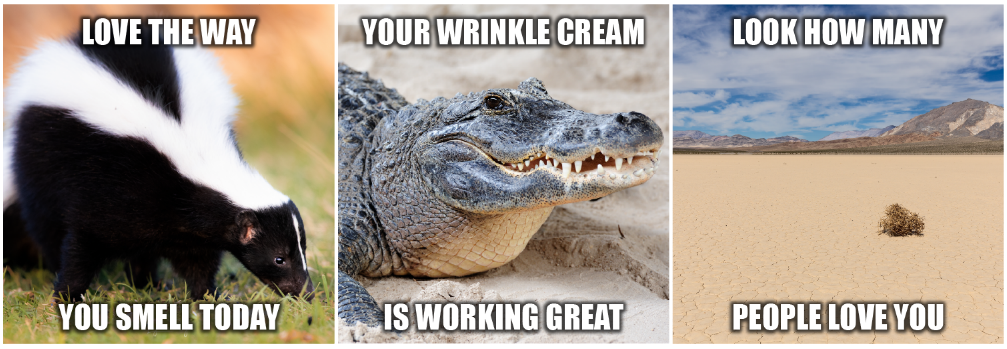

Figure 1.1: Example of memes [2]

1.2 Thesis structure

The rest of the thesis is structured as follows:

- Hateful Memes Challenge presents information related to the challenge as well as winning models in detail.
- Methodology is about theory of what have been implemented to carry out the experiment.
- Experiments shows how experiments were carried out.
- Results evaluates experiments' results and comparison with winning models.
- Conclusion and future work concludes what have been done and lists what could be done.

Chapter 2

Hateful Memes Challenge

2.1 Dataset

In this Hateful Memes dataset, memes related to being hostile toward people because of race, nationality, religion, gender, sexuality, any relevant current status (medical, mental, etc) are considered as hateful [2]. The dataset can be downloaded from `https://hatefulmemeschallenge.com/#download`. The dataset is around 4 GiB. The structure of folder is as follows:

- `LICENSE.txt`
- `README.md`
- `img/` - folder that contains around 12 000 memes as `.png` files.
- `train.jsonl`
- `dev_seen.jsonl` - development set for Phase 1.
- `test_seen.jsonl` - test set for Phase 1.
- `dev_unseen.jsonl` - development set for Phase 2.
- `test_unseen.jsonl` - test set for Phase 2.

In each `.jsonl` file, there are `id`, `img` (relative path to images), `label` (0 as non-hateful and 1 as hateful), `text` (raw strings appearing in images) keys. By counting the number of labels in each `.jsonl` except test sets, it's concluded that there are:

- 5481 labels 0, 3019 labels 1 in `train.jsonl`
- 253 labels 0, 247 labels 1 in `dev_seen.jsonl`
- 340 labels 0, 200 labels 1 in `dev_unseen.jsonl`

The next two memes are got from the dataset as instances (also across all sections). MFW means My Face When. Figure 2.1 is marked as hateful because it shows self-hatred. On the other hand, Figure 2.2 express self-love. Metadata of Meme 2.1 and 2.2 which appears in one of `.jsonl` files:

```
{"id": "10746", "img": "img/10746.png", "label": 1, "text": "mfw exist"}
{"id": "08659", "img": "img/08659.png", "label": 0, "text": "mfw exist"}
```

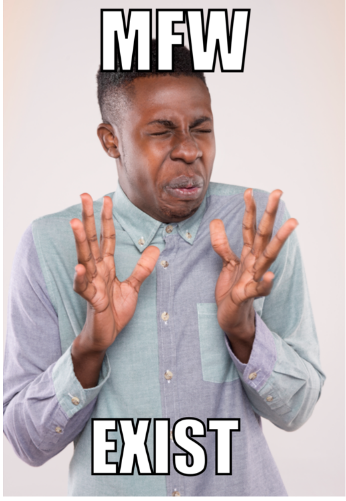

Figure 2.1: A hateful meme - File name: `10746.png`

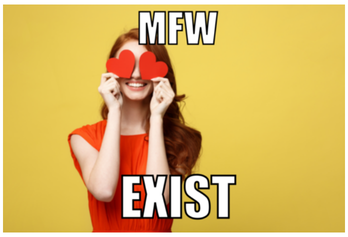

Figure 2.2: A non-hateful meme - File name: `08659.png`

2.2 Performance metrics

2.2.1 Accuracy

Accuracy is to determine how accurate a classification model's prediction is. It is calculated by the ratio of true predictions over all predictions.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Negative} + \text{False Positive}} \tag{2.1}$$

Table 2.1: Confusion matrix [3]

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

2.2.2 Area Under the Curve of the Receiver Operating Characteristic

Area Under the Curve of the Receiver Operating Characteristic (AUC ROC) is to determine how well a classification model separate between classes. In this paper's scope, after prediction, a **binary** classification model produces a list of labels with corresponding probabilities. With that list and an alterable threshold, Figure 2.3 is plotted as a sample.

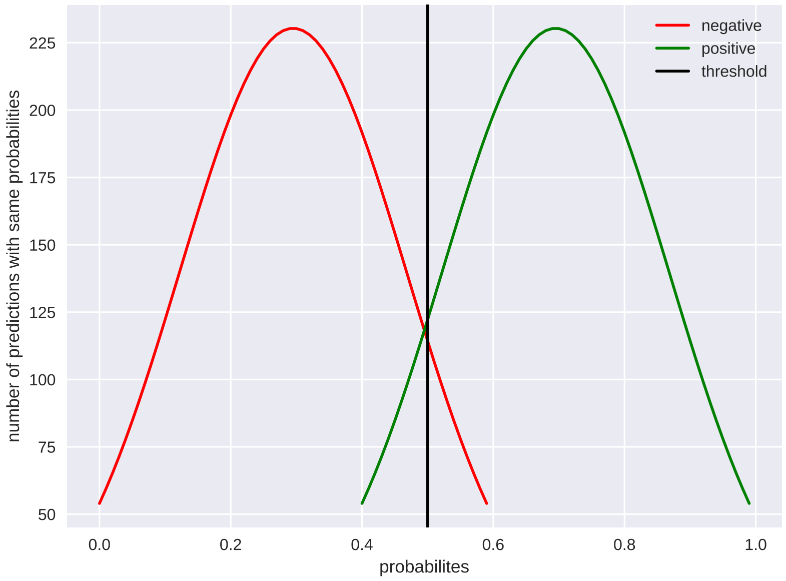

Figure 2.3: Distribution of labels with corresponding probabilities with alterable threshold

Table 2.2: Type of prediction with the threshold

	Positive predictions	Negative predictions
Above the threshold	True positive	False positive
Below the threshold	False negative	True negative

True Positive Rate (TPR) and False Positive Rate (FPR) are defined with 2.2 as following:

$$\text{TPR} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{2.2}$$

$$\text{FPR} = \frac{\text{False positive}}{\text{True negative} + \text{False positive}} \tag{2.3}$$

As the threshold sliding, TPR and FPR changes covariatedly. As a result, a curve with horizontal axis as FPR and vertical axis as TPR is plotted. This curve is called Receiver Operating Characteristic (ROC) as in Figure 2.4 sample. Area Under the Curve (AUC) is equal to the integral of the ROC from 0 to 1 in respect to TPR.

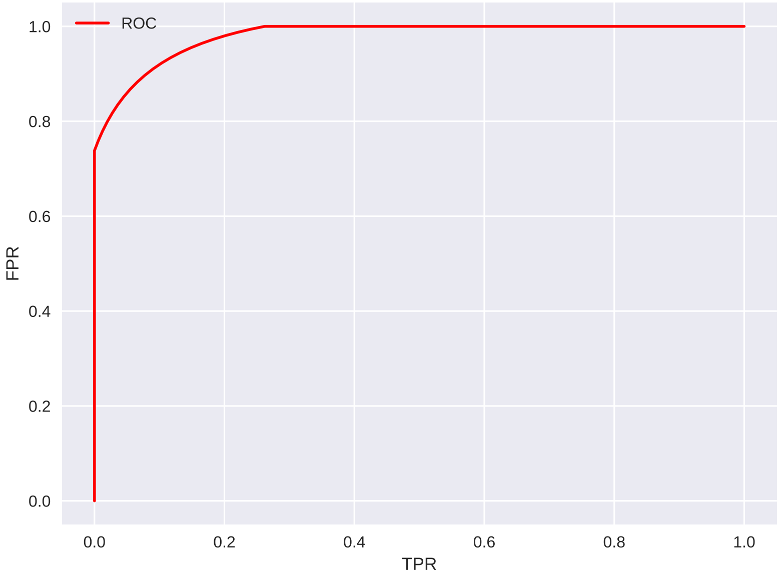

Figure 2.4: Receiver Operating Characteristic curve

2.3 Winning models

Before winning models discussion, it's good to have a short overview in MMML field. There are 3 popular problems namely representation, alignment and fusion. Representation means the way to constiute multimodal data productively. Alignment is to figure out the relationship between modalities. Fusion is attempted to combine from two or more modalities. There are two types: Early Fusion (EF) and Late Fusion (LF). EF attempts to transform all modalities at once. On the other hand, LF transforms each modality separately then fuses them later. In visual-linguistic domain, LF models are ViLBERT [4], ERNIE-ViL [5]; EF models are VL-BERT [6], VisualBERT [7], OSCAR [8] UNITER [9]. All models were created to overcome problems and solve tasks such as Visual Question

Answering (VQA) [1], Visual Commonsense Reasoning (VCR) [2] and much more. To support these model, in preprocessing step, object detection models (Faster R-CNN [10], Mask R-CNN [11]) had been reused. Moreover, general purpose Transformer[12]-based models (BERT [13], ERNIE [14]) were utilized. It's known that these models were trained on rich image datasets and meaningful corpus, for example COCO Dataset [15] and English Wikipedia corpus [3].

Winning models (5 of them), which were determined by Phase 2 results [4], had similarities in choosing models yet differences in working with data. The common V+L models were UNITER, OSCAR, ViLBERT, VL-BERT, VisualBERT, ERNIE-ViL. Image model such as Faster R-CNN and Mask R-CNN were mainly used to extract feature, produce bounding boxes, propose possible objects. As a result, V+L models had chances to know what things are in images. Text model, which are BERT, ERNIE, tokenized, encoded text. Thanks to doing so, texts were prepared carefully before fitting into V+L models. The common workflow of all wining solutions was to extract image features, to encode texts then fit to different mulitmodal models, lastly to apply ensemble learning.

There were the first place [16] and the fifth place [17] that applied data enrichment (adding more features) techniques. The first place harnessed the power of Google Vision Web Entity Detection and FairFace [18] to generate image's context as of runtime. On the other hand, the fifth place used Im2txt [19] to describe image as texts. In the other word, image captioning was carried out. As a result, V+L models had chances to know what happening in images a long with what relating to. For data expansion (adding more related memes and augmenting data), the third place [20] manually picked from Memotion Dataset while the fourth place [21] unsampled texts in images. Consequently, V+L models had more data to be trained one. Beside those data manipulation techniques, optimization method like loss re-weighting, evolutionary had been used. Moreover, ensemble algorithm namely average prediction, majority voting were implemented. In ensemle learning, models were combined with themselves but different random seeds as well as other models. Uniquely, the second place [22], they made Vilio the framework consisting of a lot of best visual-linguistic models to combine predictions.

[1] https://visualqa.org/

[2] https://visualcommonsense.com/

[3] https://www.english-corpora.org/wiki/

[4] https://hatefulmemeschallenge.com/#leaderboard

Chapter 3

Methodology

Learning from winning models mentioned in 2.3, the methodologies are complied with new changes and ideas influenced mostly by the fifth place (Section 3.1, 3.2, 3.5) and partially by the first place (Section 3.4) in the challenge. The overview of these are demonstrated in Figure 3.1 and in larger Figure .1.1 on page 32. All methodologies are described fully in regard to programming, but are not into mathematics.

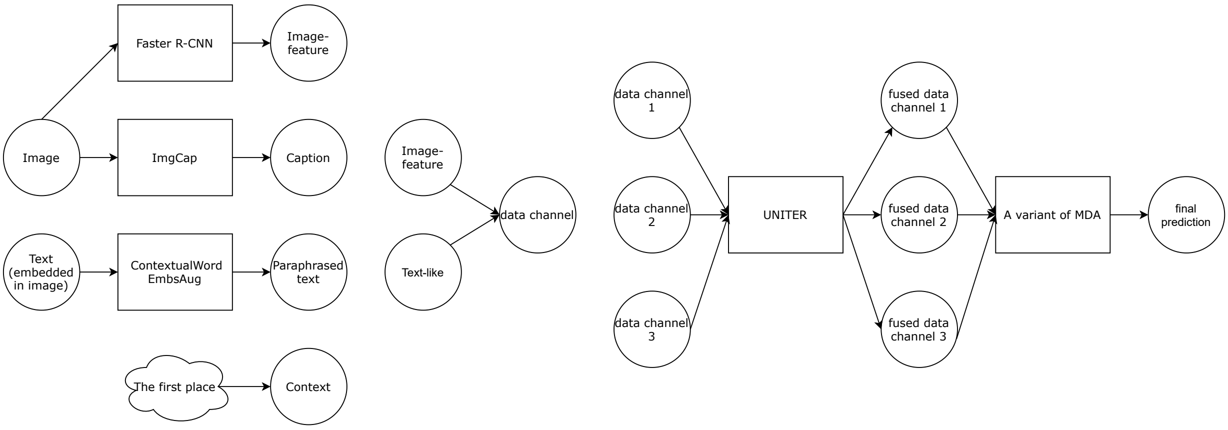

Figure 3.1: The overview of methodologies

3.1 Image feature extraction

In order not to invent the wheel, image feature (height, width, number of boxes, bounding boxes positions, feature) are retrieved from image by pre-trained Faster RCNN [23] [1] which was originally created for object detection task in computer vision. Object detection means to locate the presence of objects in image by show their bounding boxes labelled with classes. The model consists of three parts:

- A set of ResNet [24] layers - an image model to extract image feature

[1] https://github.com/airsplay/py-bottom-up-attention

- Region Proposal Network [23] - a layer to suggest bounding boxes of possible objects
- Classifier - an image model to classify possible objects.

Unlike other Faster RCNN being trained on COCO dataset, this model with ResNet (pre-trained on ImageNet [25]) was trained on VisualGenome [26]. The model would propose bounding boxes for Figure 2.1 and 2.2 as in Figure 3.2 and Figure 3.2 respectively.

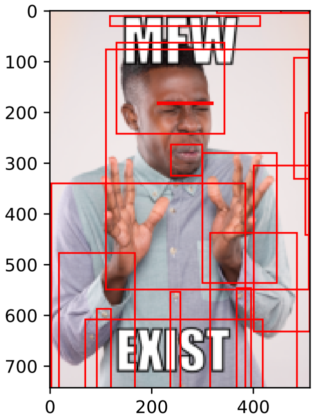

Figure 3.2: Bounding boxes of `10746.png`

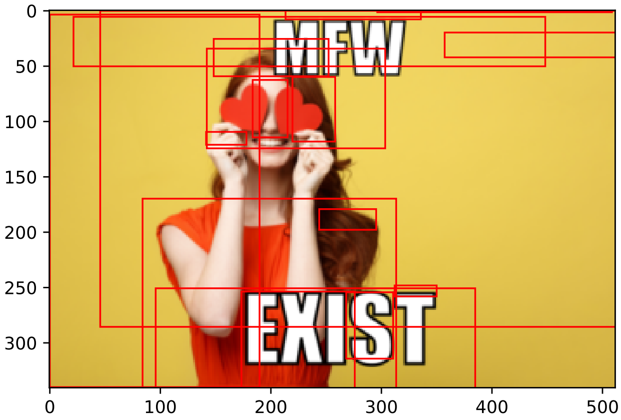

Figure 3.3: Bounding boxes of `08659.png`

3.2 Image captioning

People can understand an image by listening/reading the caption. Applying that idea to visual-linguistic models, it can help those with better understanding on what happening in the images. The captioning is the result of modality translation from vision to linguistic. To approach automation, ImgCap, which is guided by Tensorflow Tutorial [2] inspired by

[2] https://www.tensorflow.org/tutorials/text/image_captioning

”Show, Attend and Tell: Neural Image Caption Generation with Visual Attention” [27], is trusted to caption. ImgCap is simply enough to be recreated and reused for other missions. The model’s architecture can be simplified in threefold:

- Pretrained InceptionV3 [28] - an image model to extract image feature.
- Dense - merely a regular neural network to encode feature.
- GRU [29] - a language model decoding feature to generate text with Bahdanau attention [30] to focus on important feature.

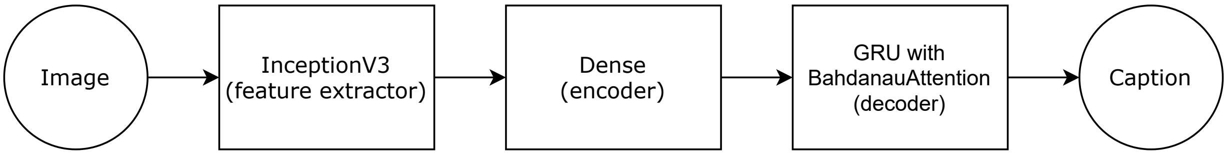
Image
InceptionV3 (feature extractor)
Dense (encoder)
GRU with BahdanauAttention (decoder)
Caption

Figure 3.4: ImgCap’s architecture

ImgCap would describe Figure 2.1 by ”a man in a sky and a tie in the inside .” and Figure 2.2 by ”a woman with a person eating something while on in one .”

3.3 Text paraphrasing

After being paraphrased, one sentence’s idea or meaning are unchanged. Therefore, a model is fitted with different sentences having one meaning, the model could perform better when encountering synonyms. Because Nlpaug[31] has good reputations and end-to-end functions, `ContextualWordEmbdsAug` from that is trusted to paraphrase text. This augmenter uses a Transformer-based model to process with two required parameters: model and action. Actions are substitute (to randomly replace word) and insert (to randomly add words). Models are bert-base-uncased, distilbert-base-uncased, roberta-base, distilroberta-base, xlnet-base-cased, bert-base-cased. ”Cased/uncased” indicates that model does/-doesn’t distinguish upper case and lower case words. As simplification, the base model are BERT [13], RoBERTa [32] (BERT with more training data), DistilRoBERTa [3] (RoBERTa with half number of parameters), XLNet [33]. For instance of this augmenter in use, using bert-base-uncased to insert could paraphrase ”mfw exist” into ”mfw also exist”. It has added one word - ”also” to the original sentence. The meaning is still unchanged.

3.4 Image context addition

For humans, to understand something, it’s required context or background knowledge. For example, when one looks at a painting, one should know what things are displayed with what attributes in order to perceive it more deeply. Therefore, the first place [16] has generated context to improve visual-linguistic model performance. The context was reused in this work for the sake of time. As illustration, the context for Figure 2.1 is:

[3] https://huggingface.co/distilroberta-base

```
{"id":"10746","img":"img\/10746.png",
"partition_description":"The Boy Envy hand"}
```

3.5 UNITER with multiple directional attention

UNITER (UNiversal Image-TExt Representation Learning) [9], which is the state-of-the-art model (in 2020) to fuse visual modality and linguistic modality, is selected for this work. UNITER learned from 4 large visual-linguistic datasets namely COCO [15], SBUCaptions [34], ConceptualCaptions [35], VisualGenome [26]. Therefore, it has quite wide large knowledge space. Pretrained UNITER$_{Large}$ and UNITER$_{Base}$ models are free to access. Large model has twice the amount of parameters as Base model. The architecture in a nutshell, UNITER is merely a stack of BERT [13] models with 2 embedders. Image embedder takes image feature, bounding boxes positions, number of boxes, imalge height, image width. Text embedder takes tokenized (by BERT) texts. Afterward, the output of 2 embedders are concatenated to fit into a stack of BERT models. As a result, a fused modality is created.

Multiple Directional Attention (MDA), which is proudly created originally, is based on bidirectional cross-attention of the fifth place [17]. Moreover, attention mechanism in use is same with MultiHeadAttention [12]. MDA allows UNITER to correlate many data channels at the same time. In this paper's scope, a data channel is a set of image and text-like. UNITER combines elements of a data channel into a single one. Then, for each pair of data channels, an attention layer is applied between them and directed from one data channel to another. In the other word, in a directed graph, data channels are considered as nodes and attention layers are treated as edges (which are directed from one node to another). Each attention layer's result goes through an attention pooling layer. Afterward, all the attention pooling layers' results are concatenated to be fit into a regular neural network. This neural network with cross entropy loss and softmax activation is able to classify binarily.

In figures 3.6, 3.7, 3.8, 3 variants of MDA namely triple cyclic attention, quadruple attention and hextuple attention are pre-defined. Furthermore, bidirectional-cross attention is a special variant of MDA when the fully-connected directed graph has two nodes. It's kept in mind that names, orders, directions are strictly relevant to 4 Experiments.

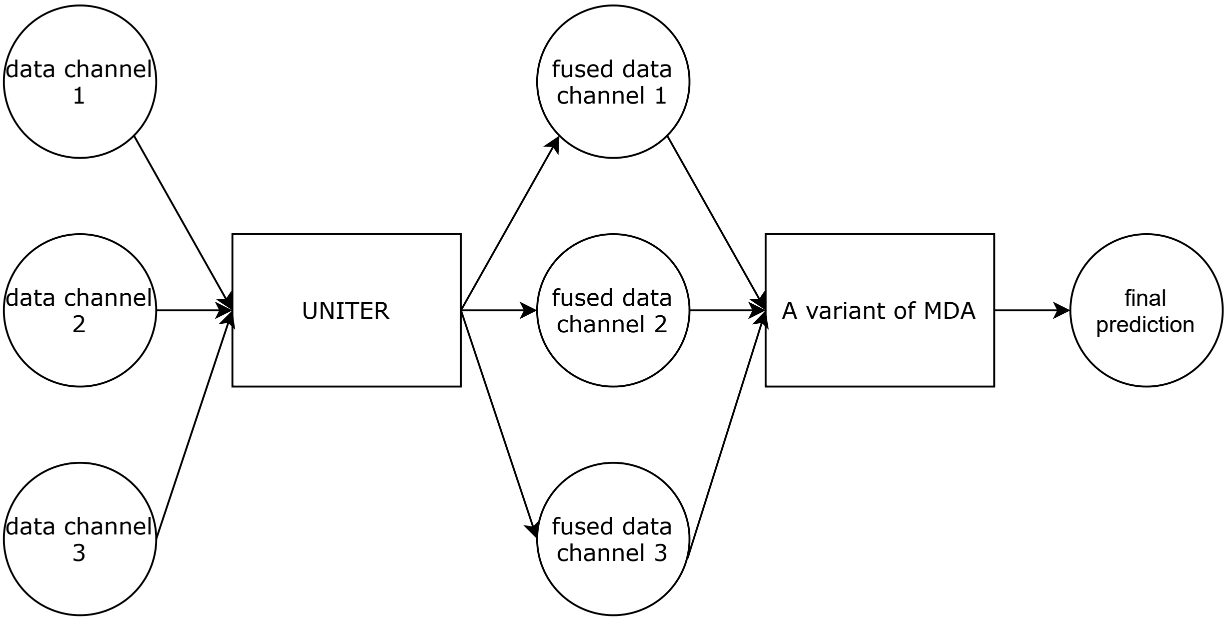

Figure 3.5: The architecture of UNITER and MDA with 3 data channels

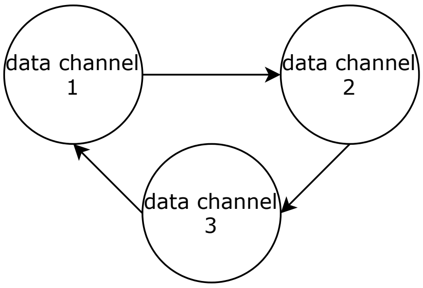

Figure 3.6: Triple cyclic attention

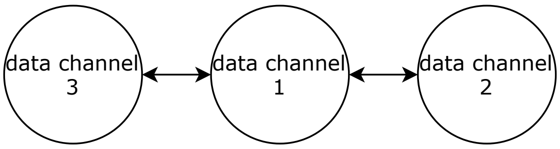

Figure 3.7: Quadruple attention

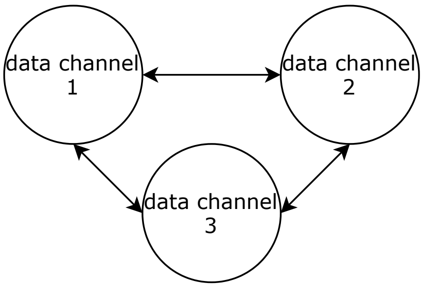

Figure 3.8: Hextuple attention

Chapter 4

Experiments

All experiments were carried out on:

- a GPU Tesla K80 12 GiBs
- a CPU Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz

It's expected that all experiments use almost 12 GiBs of RAM.

The general workflow followed the baseline of single UNITER$_{Large}$ + bidirectional cross-attention [17]. The general workflow consisted of 5 steps. From first step to fourth step, UNITER's outputs were generated. In other word, these steps pre-processed raw image and text.

1. Image height, image width, number of boxes, bounding boxes and feature were extracted from images by Faster R-CNN. The process took 3 to 4 hours.
2. With COCO 2014 dataset [15], ImgCap was trained. Next, ImgCap captioned images. It's spent 10 hours on training and almost 2 hours on captioning.
3. ContextualWordEmbsAug with different options spent 1 hours to 2 hours (depending on Transformer models and action) on paraphrasing texts (raw strings appearing in images).
4. After obtaining pre-fetched context from the first place, for each meme, all discrete contexts (tags, entities, domains) were concatenated into a single long text. It took less 5 minutes to do.
5. UNITER with a MDA variant and different inputs (generated in previous steps) was trained. It's important to know that all hyperparameters stayed the same except UNITER types, MDA variants and training inputs. Training process duration could be range from 8 hours to 10 hours regard to configurations.

In the last step, there were 4 data channels:

- image, text (always be data channel 1)
- image, caption (always be data channel 2)

- image, paraphrased text
- image, context

It's noted that image here means image height, image width, number of boxes, bounding boxes and feature.

There were 4 phases of experiments. Each phase, small changes in workflow's configuration's inputs were made.

Experiment phase 1 Find the best caption:

In this phase, ImgCap was trained 2 times to produce 2 caption files namely hm_caption.csv, hm_caption_2.csv. ContextualWordEmbsAug paraphrased texts by using `bert-base-uncased` to `insert` text. For each caption file, it's trained with followings:

a UNITER$_{Large}$ + bidirectional cross-attention (with data channel 1 and data channel 2).

b UNITER$_{Base}$ + triple cyclic attention ((image, paraphrased text) as data channel 3).

Experiment phase 2 Find the best paraphrased text:

In order to find the best options for ContextualWordsAug, 9 combinations of options were set. With best caption from the previous step, for each combination, it's trained with UNITER$_{Base}$ + triple cyclic attention ((image, paraphrased text) as data channel 3). List of those 9 options were:

a use `bert-base-uncased` to `substitute` text

b use `distilbert-base-uncased` to `insert` text

c use `distilbert-base-uncased` to `substitute` text

d use `distilroberta-base` to `insert` text

e use `distilroberta-base` to `substitute` text

f use `xlnet-base-cased` to `insert` text

g use `xlnet-base-cased` to `substitute` text

h use `bert-base-cased` to `insert` text

i use `bert-base-cased` to `substitute` text

Experiment phase 3 Find the best variant with paraphrased text:

With best caption and paraphrased text from preceding steps, UNITER$_{Base}$ was trained with MDA variants: triple cyclic attention, quadruple attention, hextuple attention.

Experiment phase 4 Find the best variant with context:

This last phase was merely almost same with previous phase except for the fact that (image, context) replaced (image, paraphrased text) to be data channel 3.

All phases' models were tested on Hateful Memes Phase 1. In addition, the last two phases' models were also tested on Hateful Memes Phase 2. Each experiment was for a single UNITER model. AUC ROC and accuracy were calculated by using scikit-learn [36], [37].

Chapter 5

Results

In this chapter, internal comparison focuses on models from experiments described in Chapter 4. On the other hand, external comparison discusses the best model with the baseline and other papers' models in the challenge.

5.1 Internal comparison

Experiment phase 1

Table 5.1: Experiment phase 1 result

	Phase 1	
Model	AUC ROC	Accuracy
hm_caption.csv#	0.7528	0.7070
hm_caption_2.csv#	0.7455	0.7190
hm_caption.csv*	**0.7845**	**0.6950**
hm_caption_2.csv*	0.7745	0.7000

#UNITER$_{Large}$ + bidirectional cross-attention
*UNITER$_{Base}$ + triple cyclic attention
((image, paraphrased text) as data channel 3)

By adding the third data channel, * models AUC ROC are higher than # models by about 0.02 to 0.03. The best model here is hm_caption.csv* even though it got the worst accuracy.

This Experiment phase 1 had proven that adding a third data channel means better performance. Moreover, different captions could lead to moderate changes in scores.

Table 5.2: Experiment phase 2 result

	Phase 1	
Model	AUC ROC	Accuracy
bert-base-uncased, insert*	**0.7845**	**0.6950**
bert-base-uncased, substitute	0.7749	0.7030
distilbert-base-uncased, insert	0.7620	0.6980
distilbert-base-uncased, substitute	0.7790	0.6930
distilroberta-base, insert	0.7750	0.6960
distilroberta-base, substitute	0.7728	0.6950
xlnet-base-cased, insert	0.7705	0.6820
xlnet-base-cased, substitute	0.7575	0.6910
bert-base-cased, insert	0.7791	0.6880
bert-base-cased, substitute	0.7715	0.6960

*equivalent with the best model from Experiment phase 1

Experiment phase 2

In Table 5.6, 9 experiments corresponding to 9 ones in Experiment phase 2 are indexed after the best model from Experiment phase 1. It's easily noted that the best model still hold its place, however its accuracy is still lower than some models in this phase.

Although paraphrased texts were from different models and different actions, they had similarities in regard to word choice and original text. Therefore, beside the best model, other models have quite same performance. This Experiment phase 2 showed that paraphrased text could make noticeable change if it is different from original text a lot.

Experiment phase 3

Table 5.3: Experiment phase 3 result

	Phase 1		Phase 2	
Model	AUC ROC	Accuracy	AUC ROC	Accuracy
Triple cyclic attention*	**0.7845**	**0.6950**	**0.7606**	**0.7220**
Quadruple attention	0.7677	0.7180	0.7605	0.7390
Hextuple attention	0.7660	0.7010	0.7696	0.7385

*equivalent with the best model from Experiment phase 2

all models are with (image, paraphrased text) as data channel 3

As it can be seen in Table 5.3, comparisons among 3 MDA variants, in regard to Phase 1, triple cyclic attention model still holds the best place. On the other hand, in aspect of

Phase 2, hextuple attention's result is the best one. In Phase 2, all values are not clearly different from each other.

Experiment phase 4

Table 5.4: Experiment phase 4 result

	Phase 1		Phase 2	
Model	AUC ROC	Accuracy	AUC ROC	Accuracy
Triple cyclic attention	0.7912	0.7040	0.7870	0.7335
Quadruple attention	**0.7984**	**0.7210**	**0.8026**	**0.7510**
Hextuple attention	0.7928	0.7040	0.7959	0.7370

all models are with (image, context) as data channel 3

Noticing in Table 5.4, quadruple attention outsmarted other models in both Phase 1 and 2. Therefore, it was entitled the best model to be compared in Section 5.2.

Experiment phase 3 and Experiment phase 4 showed that for each pair of same variant, context always aided model predict more well than paraphrased text. As explanation, context wield more valuable information than paraphrased text hold.

5.2 External comparison

Table 5.5: Comparison between the baseline and the best model

	Phase 1		Phase 2	
Model	AUC ROC	Accuracy	AUC ROC	Accuracy
5th place*	0.6830	0.7529		
Quadruple#	**0.7984**	**0.7210**	**0.8026**	**0.7510**

*single UNITER$_{Large}$ + bidirectional cross-attention [17] as the baseline model

#attention with (image, context) as data channel 3 in Experiment phase 4

In Table 5.5, the best model's AUC ROC is significantly higher than the baseline by 0.11%. To the contrary, the baseline's accuracy is more than 0.03%. The best model told apart non-hateful and hateful better (indicated by AUC ROC) but was confusing (shown by accuracy). The baseline model used only **two** data channels ((image, text) and (image, caption)) while the best model was able to make use of the third data channel (image, context). The baseline model used UNITER$_{Large}$ while the best model used UNITER$_{Base}$.

In Table 5.6, the top 5 models are sorted by Phase 2 values (AUC ROC then accuracy) and the best model is after them. In regard to Phase 1, our model outsmarted all places

Table 5.6: With other papers' models in the challenge

	Phase 1		Phase 2	
Model	AUC ROC	Accuracy	AUC ROC	Accuracy
1st place[16]	0.8460	0.7340	0.8450	0.7320
2nd place[22]	0.7825	0.6720	0.8310	0.6950
3rd place[20]	0.7805	0.7020	0.8108	0.7650
4th place[21]	0.7907	0.7180	0.8053	0.7385
5th place[17]	0.7681	0.6660	0.7943	0.7430
Quadruple*	**0.7984**	**0.7210**	**0.8026**	**0.7510**

#attention with (image, context) as data channel 3
in Experiment phase 4

except the first one in terms of AUC ROC. For Phase 2, the best model's result is higher than 5th place in every aspect. In justification of this, the best model is given with MDA which correlate many data channels (more than 5th place). Crucially, the model is **not** applied ensemble learning, therefore it performs more poorly than most others in Phase 2. Ensemble of different models or same models usually have finer prediction than a single model since they can reduce errors.

Chapter 6

Conclusion and future work

6.1 Conclusion

Toward MMML in general and Hateful Memes classification in specific, a novel MDA mechanism for UNITER has been created to adapt multiple data channels. Furthermore, experiments, which are about how captions, paraphrased texts, contexts and MDA variants can influence a visual-linguistic model's predictions, were conducted. Contexts hold more value than paraphrased texts do. The best model scored AUC ROC 0.8026 and Accuracy 0.7510 (higher than 5th place in the challenge), which is a notable achievement.

6.2 Future work

There are 6 ideas that are subjectively evaluated to be new directions for further trials and experiments.

- Try more MDA variants to find out better variants. There are 512 directed graphs with 3 labelled nodes [38].
- Try to combine MDA with other visual-linguistic models other than UNITER to find out better fusion models.
- Try to apply ensemble learning on a group of different visual-linguisti models to harness ”the wisdom of crowds”.
- Use DeOldify [1] to colourize 7% of images in Hateful Memes dataset being greyscale. Those images could be noise in the datasets. Therefore, it's better to cancel those. Colourization example can be found as in Figure 6.1 and Figure 6.2.
- Extract hateful memes from Memotion Dataset [39] and Multimodal Hate Speech (MMHS150K) [40] because more training means data better predictions.

[1] `https://github.com/jantic/DeOldify`

- Utilize OSCAR [8] to generate more meaningful captions consequently more accurate the prediction is. As of writing, OSCAR is the best model in image captioning task.

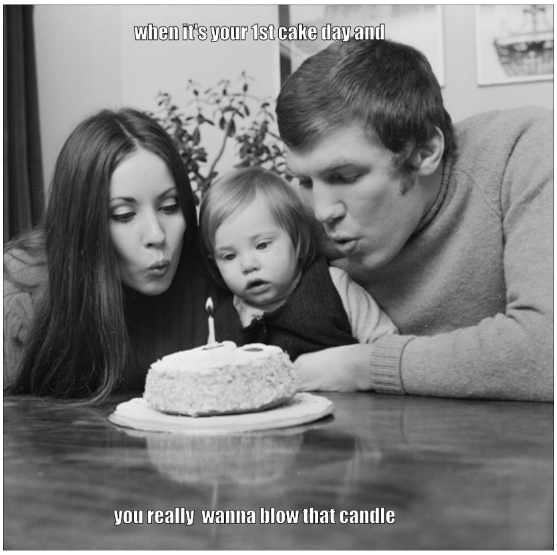
when it's your 1st cake day and
you really wanna blow that candle

Figure 6.1: A grayscale meme - File name: `53102.png`

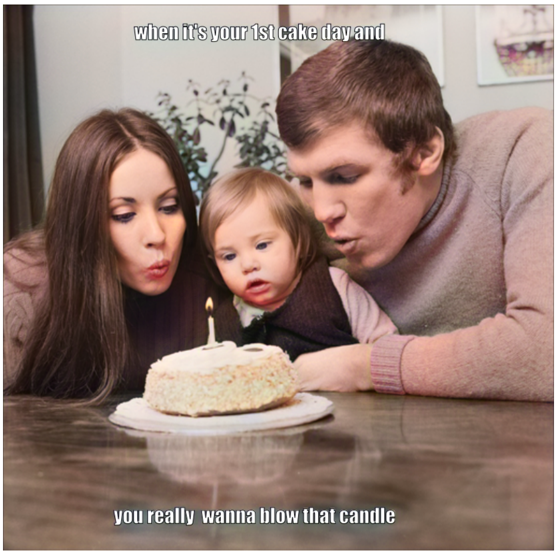
when it's your 1st cake day and
you really wanna blow that candle

Figure 6.2: Colorized Figure 6.1

Bibliography

[1] L. Börzsei, "Makes a meme instead: A concise history of internet memes," 2013.

[2] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, *The hateful memes challenge: Detecting hate speech in multimodal memes*, 2021. arXiv: 2005.04790 [cs.AI].

[3] K. Ting, "Confusion matrix," in. Jan. 2017, pp. 260–260. DOI: 10.1007/978-1-4899-7687-1_50.

[4] J. Lu, D. Batra, D. Parikh, and S. Lee, *Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*, 2019. arXiv: 1908.02265 [cs.CV].

[5] F. Yu, J. Tang, W. Yin, Y. Sun, H. Tian, H. Wu, and H. Wang, *Ernie-vil: Knowledge enhanced vision-language representations through scene graph*, 2021. arXiv: 2006.16934 [cs.CV].

[6] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, *Vl-bert: Pre-training of generic visual-linguistic representations*, 2020. arXiv: 1908.08530 [cs.CV].

[7] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, *Visualbert: A simple and performant baseline for vision and language*, 2019. arXiv: 1908.03557 [cs.CV].

[8] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao, *Oscar: Object-semantics aligned pre-training for vision-language tasks*, 2020. arXiv: 2004.06165 [cs.CV].

[9] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, *Uniter: Universal image-text representation learning*, 2020. arXiv: 1909.11740 [cs.CV].

[10] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2016. arXiv: 1506.01497 [cs.CV].

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn*, 2018. arXiv: 1703.06870 [cs.CV].

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[14] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, *Ernie: Enhanced language representation with informative entities*, 2019. arXiv: 1905.07129 [cs.CL].

[15] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV].

[16] R. Zhu, *Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution*, 2020. arXiv: 2012.08290 [cs.CL].

[17] V. Sandulescu, *Detecting hateful memes using a multimodal deep ensemble*, 2020. arXiv: 2012.13235 [cs.LG].

[18] K. Kärkkäinen and J. Joo, *Fairface: Face attribute dataset for balanced race, gender, and age*, 2019. arXiv: 1908.04913 [cs.CV].

[19] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 652–663, Apr. 2017, ISSN: 2160-9292. DOI: 10.1109/tpami.2016.2587640. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2016.2587640.

[20] R. Velioglu and J. Rose, *Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge*, 2020. arXiv: 2012.12975 [cs.AI].

[21] P. Lippe, N. Holla, S. Chandra, S. Rajamanickam, G. Antoniou, E. Shutova, and H. Yannakoudakis, *A multimodal framework for the detection of hateful memes*, 2020. arXiv: 2012.12871 [cs.CL].

[22] N. Muennighoff, *Vilio: State-of-the-art visio-linguistic models applied to hateful memes*, 2020. arXiv: 2012.07788 [cs.AI].

[23] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *Imagenet large scale visual recognition challenge*, 2015. arXiv: 1409.0575 [cs.CV].

[26] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332.

[27] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, *Show, attend and tell: Neural image caption generation with visual attention*, 2016. arXiv: 1502.03044 [cs.LG].

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, 2015. arXiv: 1512.00567 [cs.CV].

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, 2014. arXiv: 1412.3555 [cs.NE].

[30] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2016. arXiv: 1409.0473 [cs.CL].

[31] E. Ma, *Nlp augmentation*, https://github.com/makcedward/nlpaug, 2019.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL].

[33] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, *Xlnet: Generalized autoregressive pretraining for language understanding*, 2020. arXiv: 1906.08237 [cs.CL].

[34] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *Neural Information Processing Systems (NIPS)*, 2011.

[35] E. G. Ng, B. Pang, P. Sharma, and R. Soricut, "Understanding guided image captioning performance across domains," *arXiv preprint arXiv:2012.02339*, 2020.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: Experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[38] ml0105 (https://math.stackexchange.com/users/135298/ml0105), *Number of directed graphs*, Mathematics Stack Exchange, URL:https://math.stackexchange.com/q/2573788 (version: 2017-12-19). eprint: https://math.stackexchange.com/q/2573788. [Online]. Available: https://math.stackexchange.com/q/2573788.

[39] C. Sharma, D. Bhageria, W. Scott, S. PYKL, A. Das, T. Chakraborty, V. Pulabaigari, and B. Gamback, *Semeval-2020 task 8: Memotion analysis – the visuo-lingual metaphor!* 2020. arXiv: 2008.03781 [cs.CV].

[40] R. Gomez, J. Gibert, L. Gomez, and D. Karatzas, *Exploring hate speech detection in multimodal publications*, 2019. arXiv: 1910.03814 [cs.CV].

Appendices

.1 Figures

In this section, figures, which are difficult to be seen (when this paper is physically printed), are illustrated again to provide better readability.

Figure .1.1 is same with Figure 3.1 but is rotated 90 degree.

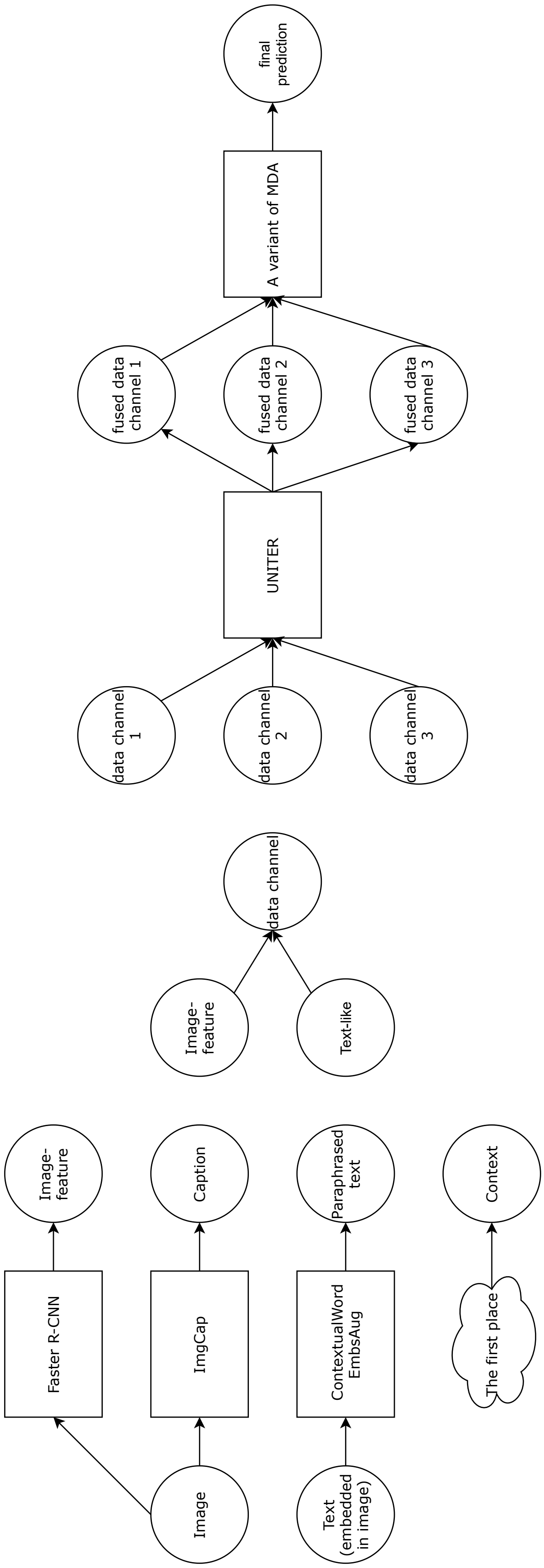

Figure .1.1: The overview of methodologies