Instagram Intent Modeling using *Meaning Multiplication*

Teerapat Jenrungrot tjenrung@cs.washington.edu
Mutita Siriruchatanon siriruc@uw.edu
Quoc Dung Cao qcao10@uw.edu

Abstract

Kruk et al. (Kruk et al., 2019) proposed a deep multimodal classifier to capture the complex relationship between image and text of Instagram posts. In this work, we successfully reproduced their main experiments. Our results supported the authors' hypotheses where a combined image and text model improves the classification of intent, contextual, and semiotic relationships compared to single modality models, and ELMo contextual language embedding outperforms word2vec, a traditional non-contextual word embedding. Furthermore, we experimented on CBOW, a variant of word2vec embedding, and utilizing emoji in the Instagram captions. However, the classification performance remained the same for the CBOW and emoji experiments.

1 Introduction

Multimodal relationship between, for example, text and image is complex. Image caption does not necessarily have to express literal meaning of the image and vice versa. In their work (Kruk et al., 2019), Kruk et al. introduce a multimodal dataset of 1299 Instagram posts labeled with three orthogonal taxonomies: intent behind the image-caption pair, contextual relationship between the literal meanings of the image and caption, and the semiotic relationship between the signified meanings of the image and caption. In showing the relationship between two modalities, Kruk et al. use a deep convolutional neural network to combine visual representation and textual representation of Instagram posts. They claim that multimodal embeddings have better performance for intent classification than embeddings from a single modality.

Based on (Bateman, 2014) and (Kruk et al., 2019), the taxonomies that capture the relationship between text and image and their combination are **authorial intent**, **contextual relationship**, and **semiotic relationship**. The authorial intent, describing the intent behind the image-caption pair, is divided into provocative, informative, advocative, entertainment, expositive, expressive, and promotive. The contextual relationship measures the relationship between literal meanings of the image and caption. There are three groups of contextual relationship: minimal, close, and transcendent. Lastly, the semiotic relationship captures the relationship between the signified meanings of the image and caption. There are three groups of semiotic relationship: divergent, additive, and parallel. Note that the contextual taxonomy does not deal with the more complex forms of "meaning multiplication." A *divergent* relationship happens when the image semiotic and caption semiotic are in opposite directions. A *parallel* relationship happens when the image and caption semiotics independently contribute to the same meaning. An *additive* relationship happens when the image semiotic or caption semiotic amplifies the other.

2 Contributions

The main contributions for this paper are (1) reproducing the results obtained by (Kruk et al., 2019) and (2) offering additional insights regarding image-text relationship modeling through other text embedding networks, and utilizing emoji in the caption text.

2.1 Hypotheses from original paper

According to the authors, image and text contributions to authors' intent are far from asymmetrical, i.e one is primary and one is complementary. Furthermore, the meaning should not be additional. Determining author intent with text-image content requires a richer kind of meaning composition that has been referred to by the authors as *meaning multiplication* (Bateman, 2014). The hypotheses from the original paper are listed as follow:

- Hypothesis 1: Combining both text and image improves a detection of authorial intent/contextual relationship/semiotic relationship compared to using only one modality

- Hypothesis 2: Using pre-trained ELMo embedding would have higher accuracy than using word2vec embedding trained from scratch

2.2 Hypotheses addressed in this work

The hypotheses 1 and 2 are similar to the original work, serving to study the reproducibility of the paper. The remaining hypotheses are our additional experiments.

- Hypothesis 1: Combining both text and image improves a detection of authorial intent/contextual relationship/semiotic relationship compared to using only one modality

- Hypothesis 2: Using pre-trained ELMo embedding would have higher accuracy than using word2vec embedding trained from scratch

- Hypothesis 3: Using a continuous bag of words (CBOW) embedding trained from scratch would have higher accuracy than using word2vec embedding trained from scratch

- Hypothesis 4: Utilizing emoji in the caption text would provide higher accuracy than the processed caption without emoji by the original paper (Kruk et al., 2019)

2.3 Experiments

All experiments used a pre-trained ResNet-18 as an image-encoder and default hyperparameters as shown in Table 2. The list of experiments are presented in Table 1.

3 Code

All implementations were done using PyTorch. The code, including the script for reproducing the experiments, is publicly available at https://github.com/mjenrungrot/cse517_projects.

4 Experimental setup and results

4.1 Model description

The models presented in (Kruk et al., 2019) are based on deep convolutional neural network. The model consists of an image encoder, a caption encoder, a fusion layer, and a class prediction layer. For an image encoder, the ResNet-18 network pre-trained on ImageNet is used. For a caption encoder, the word2vec trained from scratch with 300 dimensional vectors. For pre-trained character-based contextual embeddings (ELMo), the original paper uses 2 layers of embedding resulting in a 2048 dimensional embedding. In our experiment, we used a similar pre-trained API with one layers resulting in a 1024 dimensional input. A fusion layer has a common embedding dimension of 128. The RNN model is a bidirectional GRU with a hidden size of 256. Lastly, the network is trained to optimize with the Adam optimizer using the cross-entropy loss.

4.2 Datasets

The dataset consists of 1299 posts from Instagram, including captions, tags, number of likes, the image features, and three labels, namely, 1) the authorial intent behind the image-caption pair, 2) the contextual relationship between the literal meanings of the image and caption, and 3) the semiotic relationship between the signified meanings of the image and caption. The number of classes of 'intent' relationship, 'contextual relationship' relationship, and 'semiotic' relationship are 7, 3, and 3, respectively. The labels counts by the paper authors show that the dataset is imbalanced as the majority of the labels in each taxonomy belongs to one or two classes.

The data can be accessed through https://github.com/karansikka1/documentIntent_emnlp19. The provided data was split into 5 sets for 5-fold cross-validation for their experiment. Due to restriction on sharing contents from social media sites such as Instagram, the authors only share the featurized images from the ResNet-18 architecture (He et al., 2016) for the purpose of reproducibility study and further experimentation. For this reason, we do not update the weights corresponding to an image encoder in all of our experiments.

Experiment	Description	Results	Hypotheses Supported
Experiment 1	Run the model with a single modality and a combined Img + Txt as inputs using word2vec/ELMo as a caption encoder	Table 3	Hypothesis 1 and 2
Experiment 2	Run the model with Txt only and a combined Img + Txt as inputs using CBOW as a caption encoder	Table 4	Hypothesis 3
Experiment 3	Create a new caption from the original caption by retaining emoji and repeat Experiment 1	Table 5	Hypothesis 4

Table 1: Description of experiments

Hyperparameter	Value
number of epoch	70
batch size	4
word2vec	300 dim
CBOW - size of context window	2
pre-trained ELMo	1024 dim
fusion layer	128 dim
biderectional GRU	256 dim
initial learning rate	0.00005
learning rate scheduler	step decay
learning rate scheduler patience	15 epochs
learning rate scheduler reduction factor	0.1
learning rate optimizer	Adam

Table 2: Default hyperparameters. All other unmentioned hyperparameters are PyTorch default.

4.3 Hyperparameters

Table 2 describes all important hyperparameters the network requires. All other unmentioned hyperparameters are PyTorch default hyperparameters.

4.4 Results

We first compared our reproduced results (Table 3) with the original results. Considering all methods and taxonomies, the absolute differences in ACC are within the range of 0 to 0.06, and the ranges for AUC are 0.0008 to 0.05. Due to small deviations from the original results, we conclude that we successfully reproduce the experiments in the original paper.

Generally, combining two modalities improves the performances for all taxonomies (Table 3). In most cases, adding images improves both ACC and AUC when word2vec embedding is used. However, a combined modalities with ELMo performs similarly to a single modality with ELMo. Still, the winners for all taxonomies are Img + Txt regardless of metrics with the exception of semiotic taxonomy with ACC as metrics. Consequently, the results of Experiment 1 support the claim in Hypothesis 1.

As shown in Table 3, encoding caption with ELMo improves both ACC and AUC compared to word2vec embedding, except for the semiotic taxonomy. In semiotic taxonomy, the performances of Txt-ELMo and Txt-emb are similar, which is consistent with the result in the original paper. Therefore, our results support the claim in Hypothesis 2 for intent and contextual taxonomies. This implies we only require individual words to predict the semiotic taxonomy. We also noted from the results in (Table 3), when trying to replicate the original results, that using one layer of ELMo embedding yields almost the same performance as using two layers of embedding in the original paper.

Method	Intent		Semiotic		Contextual	
	ACC	AUC	ACC	AUC	ACC	AUC
Img	0.3857	0.7398	0.6258	0.5987	0.5296	0.6159
Txt-emb	0.4149	0.7121	0.6220	0.6442	0.5812	0.6935
Txt-ELMo	0.5420	0.8292	0.6144	0.6380	0.6328	0.7522
Img + Txt-emb	0.5473	0.7861	0.6166	0.6578	0.5866	0.7186
Img + Txt-ELMo	0.5462	0.8286	0.6122	0.6314	0.6360	0.7628

Table 3: Results for Experiment 1. For this experiment, we run the model with a single modality and a combined Img + Txt as inputs using word2vec/ELMo as a caption encoder. This experiment is the reproduced experiment from (Kruk et al., 2019).

5 Experiments beyond the original paper

Our first additional experiment (Experiment 2) aims to answer the question of how well different models work on modeling image-text pairs. In this experiment, we use a continuous bag-of-word

Method	Intent		Semiotic		Contextual	
	ACC	AUC	ACC	AUC	ACC	AUC
Txt-CBOW	0.4334	0.7000	0.6466	0.5838	0.6067	0.7101
Img + Txt-CBOW	0.4465	0.7750	0.6436	0.6217	0.5959	0.7002

Table 4: Results for Experiment 2. For this experiment, we run the model with Txt only and a combined Img + Txt as inputs using CBOW as a caption encoder.

model (Mikolov et al., 2013) to embed the provided captions. To embed each word, CBOW also considers the surrounding context of the word. We set the size of context window as 2. This means that for every word, we will consider 2 words before and after in additional to the word itself.

Additionally, we attempt to answer the caption whether emojis in Instagram captions significantly improve model performance. In the original papers, the authors disregarded all emojis during the preprocessing step. However, we hypothesize that emojis are important features for estimating authors' intention. To create a new caption, we used the raw caption provided by the authors and processed using the following rules.

1. Remove all apostrophe contractions
2. Remove any hashtags
3. Remove any tags at the end of the post while consider tags appeared in the middle of the post as part of a caption
4. Remove stopwords and non-alphanumeric characters that are not emoji
5. Add space between emojis if there is none

Note that the authors briefly mentioned how they preprocessed the captions. However, we found out that they used rule 1, 2, and 3 in their pre-processing and removed all non-alphanumeric characters. We used Python packages `emoji`[1] and `emoji_data_python`[2] to check whether a word is emoji or not.

5.1 Results for additional experiments

We compared the results of text-only models (Txt-emb vs. Txt-CBOW) and joint models (Img+Txt-emb vs. Img+Txt-CBOW) from Experiment 1 and 2 (Table 3 and 4) using two metrics, ACC and AUC. However, the two metrics provide contradicting results. Since our data labels for all taxonomies are class-skewed, we solely draw a conclusion using results by AUC. CBOW embedding improves AUC in the text-only model for contextual taxonomy. This implies that surrounding context of the word only helps when the task is explaining the image-text relationship. Thus, results from Experiment 2 support Hypothesis 3 only on the task of labelling contextual taxonomy in the text-only model.

For the last hypothesis, we compared the results of each respective model from Experiment 3 (Table 5) to those of Experiment 1 and 2 (Table 3 and 4) such as Txt-emb vs. $\text{Txt}_{\text{Emoji}}$-emb, Img + Txt-ELMo vs. Img + $\text{Txt}_{\text{Emoji}}$-ELMo, and etc. There is no difference between the performance of models using captions with and without emoji for word2vec and CBOW embedding. However, including emoji to a single/joint model improves the performances in predicting the semiotic taxonomy when ELMo embedding is used (0.64 for Txt-ELMo vs. 0.67 for $\text{Txt}_{\text{Emoji}}$-ELMo and 0.63 for Img + Txt-ELMo vs. 0.68 for Img + $\text{Txt}_{\text{Emoji}}$-ELMo). Thus, the claim of Hypothesis 4 was true only for semiotic labeling task.

Method	Intent		Semiotic		Contextual	
	ACC	AUC	ACC	AUC	ACC	AUC
$\text{Txt}_{\text{Emoji}}$-emb	0.4088	0.7248	0.6236	0.6495	0.5674	0.6825
$\text{Txt}_{\text{Emoji}}$-CBOW	0.4527	0.7126	0.6474	0.5918	0.6385	0.7231
$\text{Txt}_{\text{Emoji}}$-ELMo	0.5428	0.8338	0.5972	0.6666	0.6476	0.7586
Img + $\text{Txt}_{\text{Emoji}}$-emb	0.5135	0.8030	0.6243	0.6556	0.5920	0.7292
Img + $\text{Txt}_{\text{Emoji}}$-CBOW	0.4442	0.7720	0.6297	0.6262	0.5820	0.7045
Img + $\text{Txt}_{\text{Emoji}}$-ELMo	0.5400	0.8262	0.6082	0.6778	0.6406	0.7724

Table 5: Results for Experiment 3. For this experiment, we create a new caption from the original caption by retaining emoji and repeat Experiment 1.

6 Computational requirements

The original paper did not report the type of hardware they used or the average runtime. Due to a small dataset, we expect that the computation is feasible within GPU nodes offered by the class. We run all experiments on CentOS 7.7 (64-bit Linux) by training and validating our model using 5-fold cross-validation for 70 training epochs. The number of GPU hours for each experiment are shown in Table 6, 7, and 8. In total, we spent 275 GPU hours to run all experiments. Note that any methods using ELMo as a caption encoder require 35 GPU hours, while the remaining methods only require 15 GPU hours approximately. Al-

[1] https://pypi.org/project/emoji/
[2] https://pypi.org/project/emoji-data-python/

though not implemented, we want to highlight a potential speedup when using ELMo embedding, instead of running the embedding again every time we run on another splits, or another classification type, we can run the ELMo embedding once, save the embedded caption, and load it once we need them. Overall, we only need to run the bidirectional LSTM network in ELMo through the caption dataset 5 times.

Method	GPU hours
Img	15
Txt-emb	15
Txt-ELMo	35
Img + Txt-emb	15
Img + Txt-ELMo	35
Total	115

Table 6: Number of GPU hours for Experiment 1

Method	GPU hours
Txt-CBOW	15
Img + Txt-CBOW	15
Total	30

Table 7: Number of GPU hours for Experiment 2

Method	GPU hours
$\text{Txt}_{\text{Emoji}}$-emb	15
$\text{Txt}_{\text{Emoji}}$-CBOW	15
$\text{Txt}_{\text{Emoji}}$-ELMo	35
Img + $\text{Txt}_{\text{Emoji}}$-emb	15
Img + $\text{Txt}_{\text{Emoji}}$-CBOW	15
Img + $\text{Txt}_{\text{Emoji}}$-ELMo	35
Total	130

Table 8: Number of GPU hours for Experiment 3

7 Discussion and recommendations

In this work, we reproduce the results from (Kruk et al., 2019). Based on the reproduced results, we were able to support two following hypotheses: that combining both text and image improves the task of authorial intent/contextual relationship/semiotic relationship as compared to using only one modality and that using pre-train ELMo embedding (Peters et al., 2018) has better performance than using word2vec embedding trained from scratch.

Based on our additional experiments, we show that using a variant of word2vec that incorporates contextual information, a continuous bag of word model, does not generally have higher performance than the simple word2vec embedding when trained from scratch. Additionally, including emoji in the model is shown to not improve the classification performance.

Because the dataset doesn't provide raw image features, we aren't able to investigate further on how these two modalities interact with each other. We believe that potential future research direction of multi-modal social media content processing could be beneficial from looking how the semantic of an image and semantic of a caption inter-plays.

References

John Bateman. 2014. *Text and image: A critical introduction to the visual/verbal divide*. Routledge.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, and Ajay Divakaran. 2019. Integrating text and image: Determining multimodal document intent in Instagram posts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4621–4631, Hong Kong, China. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.