NYK-MS: A Well-annotated Multi-modal Metaphor and Sarcasm Understanding Benchmark on Cartoon-Caption Dataset

Anonymous ACL submission

Abstract

Metaphor and sarcasm are common figurative expressions in people's communication, especially on the Internet or the memes popular among teenagers. We create a new benchmark named NYK-MS (NewYorKer for Metaphor and Sarcasm), which contains 1,583 samples for metaphor understanding tasks and 1,578 samples for sarcasm understanding tasks. These tasks include whether it contains metaphor/sarcasm, which word or object contains metaphor/sarcasm, what does it satirize and why does it contains metaphor/sarcasm, all of the 7 tasks are well-annotated by at least 3 annotators. We annotate the dataset for several rounds to improve the consistency and quality, and use GUI and GPT-4V to raise our efficiency. Based on the benchmark, we conduct plenty of experiments. In the zero-shot experiments, we show that Large Language Models (LLM) and Large Multi-modal Models (LMM) can't do classification task well, and as the scale increases, the performance on other 5 tasks improves. In the experiments on traditional pre-train models, we show the enhancement with augment and alignment methods, which prove our benchmark is consistent with previous dataset and requires the model to understand both of the two modalities.

1 Introduction

People often express their idea with metaphor or sarcasm. For example, they may use aliases of some important people instead of their real name, which contain metaphor and can escape from being blocked by many social networks. Besides, when they want to show their strong negative sentiment on some events, they can use words contains positive sentiment and satirize it. On the Internet, these phenomenon is becoming more and more common, bringing huge challenge for models to understand people's real meaning.

Formally, metaphor means that a word or phrase's real meaning isn't same with its basic meaning (Lakoff and Johnson, 2008; Lagerwerf and Meijers, 2008). In linguistics studies, researchers use MIP (Group, 2007; Steen et al., 2010) and SPV (Wilks, 1975, 1978) theories to define metaphor, which also provide strategies for auto detection (Choi et al., 2021). Similarly, sarcasm can be defined as that a word or phrase's real sentiment doesn't consist with its basic sentiment (Zhang et al., 2024), and usually means using positive words to express negative sentiment. In multimodal situations, the word and phrase could also be an object in the image.

To deal with the task of metaphor and sarcasm understanding, a well-annotated dataset is quite important. However, previous datasets have their weakness, such as annotated by mechanical rules (Cai et al., 2019), mainly focusing on texts or only using memes (Zhang et al., 2021). In this paper, we create a new benchmark called NYK-MS, which can benefit research on these tasks. Our work can be described through the questions and answers below:

Q1: Why do we use cartoon-caption as origin dataset?

In our early work, we tried to collect data from Twitter, but the Tweets don't meet our requirement. First, the ratio that a Tweet contains metaphor and sarcasm is very low (see Table 1). There is less than 20% samples contains metaphor, even if we use `#metaphor` as crawling tag. Besides, using selected MVSA dataset (Niu et al., 2016) is a worse choice, where only about 13% samples contains metaphor. Detailed crawling and selection method can be seen in Appendix A.

Second, the metaphor and sarcasm in normal Tweets are quite easy, and the image is not necessary. Figure 1 shows a metaphor example. In this example, even if there is no photo, we can easily judge that the word "bombed" contains metaphor. Most of the positive examples in early annotation

only contains single-modal metaphor or sarcasm, so they are not suitable with the multi-modal task.

Data Source	Pos.	Neg.	Total	Pos. Rate
Twitter	93	376	469	19.83%
MVSA	522	3476	4000	13.05%
All	615	3854	4469	13.76%

Table 1: Early annotate results. Twitter means that the data is crawled from Twitter, and MVSA means that the data is selected from MVSA dataset (Niu et al., 2016).

Figure 1: An example in early annotation that contains metaphor. The text is "Photo bombed by a deer, what an asshole."

Third, limited by crawling settings, the crawled data always focus on specific events. For example, in MVSA dataset, a plenty of samples are related with `#NationalDogDay` tag, so there are many positive metaphor words about dogs, such as "baby", "friend", "lover". However, our work doesn't focus on these topics, and we want to create a general dataset.

With these weakness, we finally give up using Tweets or MVSA dataset for annotation. Instead of these real corpus, `newyorker_caption_contest` (Hessel et al., 2023) is a new dataset contains cartoon-caption pairs. The dataset is derived from Jain et al. (2020), Shahaf et al. (2015), and Radev et al. (2016). It consists of 3 tasks: **Matching**, **Ranking** and **Explanation**. Since the cartoons are created by professional painters and always contain deep meaning, the ratio that they contains metaphor and sarcasm is far higher than the data shown above. Besides, the captions are selected from readers' submission, so the winners are always humor and concise, which requires the model must combine the two modals to understand the whole meaning, which is more difficult than Tweets. Finally, the cartoons are published in past several years, so they don't focus on specific topics. With these advantages, we choose this dataset as our origin data.

Q2: What tasks does NYK-MS support?

Our NYK-MS dataset can be separated as NYK-M and NYK-S, supporting metaphor and sarcasm task respectively.

For metaphor understanding, NYK-M contains 3 tasks:

- Metaphor Classification (**MC**). Models should output 1 or 0, representing whether the sample contains metaphor.
- Metaphor Word detection (**MW**). Models should output the word (or phrase, object in image) that contains metaphor.
- Metaphor Explanation (**ME**). Models should output the explanation of the metaphor, including the literal meaning, the real meaning and the reason of such usage.

For sarcasm understanding, NYK-S contains 4 tasks. Besides the similar task **SC** (Sarcasm Classification), **SW** (Sarcasm Word detection), **SE** (Sarcasm Explanation), we designed another task called **ST** (Sarcasm Target detection), which requires models to output the target of sarcasm, such as social phenomena or people. This task is important for application, because it can help models understand the position of speakers.

In conclusion, NYK-MS contains 7 tasks: MC, MW, ME, SC, SW, ST, SE. The detailed annotation workflow for these tasks is described in Section 3.

Q3: What experiments do we conduct?

For zero-shot situation, we use several LLMs and LMMs to do these task, and analyze their performance; For fine-tuning situation, we designed a baseline model following Zhang et al. (2024) for MC and SC task, and using several alignment methods to improve the classification rate, including contrastive learning, Optimal Transport (Villani et al., 2009). Besides, we use data augmentation and knowledge augmentation method and get improvement, showing that our annotation standard is consistent with previous work and human understanding.

Our contributions are as follow:

- We create a new well-annotated benchmark NYK-MS for multi-modal metaphor and sar-

casm understanding, including 7 tasks on more than 1,500 cartoon-caption pairs.

- We design a workflow to deal with the inconsistency of annotators, and use LMM's output as base annotation, changing the free writing task into checking and modifying task. This workflow can be used in any difficult task in which people's idea is not same.

- We conduct plenty of experiments, showing the performance of models, and use several methods to improve the performance on our NYK-MS dataset, including alignment and augmentation.

2 Related Work

2.1 Previous Datasets

For text-only metaphor understanding task, VUA18 (Leong et al., 2018), VUA20 (Leong et al., 2020), MOH-X (Mohammad et al., 2016), TroFi (Birke and Sarkar, 2006) are commonly used datasets. VUA series are annotated on word level, tagging whether each word contains metaphor. For cross-modal task, MVSA (Niu et al., 2016) is a sentiment analysis dataset, setting -1, 0, 1 three sentiment and consists of MVSA_Single and MVSA_Multiple, the latter remains three origin annotation result. HFM (Cai et al., 2019) is a large sarcasm detection dataset, but the train set is annotated by the tag `#sarcasm`, which means if the text contains the tag, the result is 1, otherwise is 0. MET-Meme (Xu et al., 2022) is a detailed dataset on meme, contains plenty of annotation items. MultiMET (Zhang et al., 2021) and MetaCLUE (Akula et al., 2023) are also multi-modal metaphor understanding datasets.

2.2 Contrastive Learning

SimCLR (Chen et al., 2020) and MoCo series (He et al., 2020) use contrastive in CV, and SimCSE uses it in NLP. CLIP (Radford et al., 2021) does alignment with contrastive learning, and ITC (Image-Text Contrastive) loss is used widely in recent multi-modal pre-training models, including BLIP (Li et al., 2022, 2023).

2.3 Optimal Transport

OT (Optimal Transport) is a method to calculate a best transfer matrix between two distributions (Villani et al., 2009). Cuturi (2013) use an iteration method (Knight, 2008) to solve it effectively. Since the cross-modal alignment can be seen as a transfer process, Solving OT to calculate the transfer cost as loss function is used by some works (Pramanick et al., 2022; Xu and Chen, 2023; Aslam et al., 2024).

3 Data Annotation Workflow

3.1 Data Pre-process

We download the `newyorker_caption_contest` dataset (Hessel et al., 2023) from GitHub[1]. As mentioned above, the dataset contains 3 tasks:

- **Matching**. Giving a cartoon and 5 captions of different cartoons, models should choose the only related caption. This task is a 5-choice question, so only the right answer is useful for our annotation.

- **Ranking**. Giving a cartoon and 2 captions of it, models should choose the better caption. In this task, both of the 2 answers can be used for our annotation.

- **Explanation**. Giving a cartoon and its caption, models should output the explanation of them. The task can be used for our annotation, but the number of it is quite small.

We put all the images into a folder, and then analyze the JSON files. Table 2 shows the result of the 3 files for each task after data retrieval and deduplication:

Task	Cartoons	Captions	Cap. / Car.
Matching	704	2653	3.77 (Average)
Ranking	679	5127	7.54 (Average)
Explanation	651	651	1

Table 2: Analysis result for origin data. We use only the correct answer in Matching task, and the two answers in Ranking task. Cap. / Car. means the number of captions for each cartoon, and in Matching and Ranking task, the number is not fixed.

Besides, we get the union of 3 tasks, and the full set contains 704 cartoons and 5200 captions. We can see that the Ranking task contains 679 cartoons and 5127 captions, almost covering the full set, so we use the data in Ranking task for annotation. We use the 679 cartoons, and selected 3 captions randomly for each cartoon, so the origin dataset contains $679 \times 3 = 2037$ samples. For each sample, we use the cartoon, the caption and the description from the origin dataset for next step.

[1] See https://github.com/jmhessel/caption_contest_corpus.

3.2 Annotation GUI

We designed two versions of GUI using Tkinter library in Python. The first version is used in classification step, and the second version is used in modifying step. Fig 2 shows the GUI, and the details about them are shown in Appendix B.

3.3 Classification Step

With the processed data and GUI, we hired 12 annotators to do this step. These annotators are separated into 4 group, each group annotates $679/4 = 170$ (the last group is 169) cartoons. In this step, we regard the cartoon and the 3 captions as a whole, and only require the annotators to judge whether them contains deep meaning, no matter it is metaphor or sarcasm. However, after the annotation, we find that the consistency is very low. Table 3 shows the result, and the consistency in round 1 is unacceptable. Almost all pair-wise Kappa is less than 0.2, meaning no consistency at all in statistic.

So, we discuss online to deal with the consistency problem. We selected 2 inconsistency (001 or 011) samples from each group's annotation result, and invite the annotator who think it is positive to explain his opinion. More details about this discussion can be seen in Appendix D.

After the discussion, annotators start the second round annotation. For the 000 and 111 samples, we use the result from round 1, and only annotate 001 and 011 samples, which contains of $184 + 241 = 425$ samples. After this round, the consistency becomes higher, and we use the 011 and 111 samples ($123 + 259 = 382$ cartoons, $382 \times 3 = 1146$ cartoon-caption pairs) as positive samples for next step.

3.4 GPT-4V Annotation Step

With the 1146 positive samples, we use GPT-4V (through the API provided by Close-AI[2]) to generate base annotations. Table 4 shows the prompt we use.

The GPT-4V result is already formatted, answering each question in one line and starts with the number. We do these post-process:

- If the number of non-empty lines is less than 4, adding blank lines as padding;
- delete the question number at the front of each lines, such as "1. ", "2. ", $\cdots$
- Comparing the first line with word "Yes", if they are matching, setting the answer of MC/SC task as 1, otherwise setting it as 0, and set other results as empty strings.
- If the second line contains quotation marks, we selected the words between them as the answer for MW/SW task, otherwise we use the whole line as answer.
- for metaphor task, we concatenate the 3rd and 4th lines as the answer for ME; for sarcasm task, the final two lines are the answer for ST and SE respectively.

On the classification task, GPT-4V's annotation is shown in Table 5.

3.5 Checking and Modifying Step

We hired 9 annotators to check the GPT-4V result. We deleted the samples that GPT-4V answers "I'm sorry, but I can't provide assistance with that content." or gives obviously wrong answers. Then, we add the negative samples from the classification step (totally $(140 + 157) \times 3 = 891$ negative samples), and get the preliminary version of NYK-MS.

Finally, we hired 3 professional annotators to modify the annotation result. We use the samples selected in previous discussion to unify the standard and make clear and strict annotation rules. We listed some rules in Appendix C.

After this step, the building workflow of NYK-MS is finished. Table 6 and 7 shows the information of it.

Figure 3 is a metaphor sample and a sarcasm sample in NYK-MS.

4 Experiments

4.1 Zero-shot Large Model Experiments

We conduct experiments on 4 large models:

- **LLaVA** (Liu et al., 2024). LLaVA projects the embedding of image into text embedding space, then concatenate the two modalities' vector for downstream task. We use the 7B version[3].
- **GPT-3.5** (Ouyang et al., 2022). GPT-3.5 uses Pre-train, SFT, RM and PPO and achieves high conversation ability. However, it is a text-only model.

[2]See https://www.closeai-asia.com.

[3]See https://https://huggingface.co//llava-hf/llava-v1.6-mistral-7b-hf.

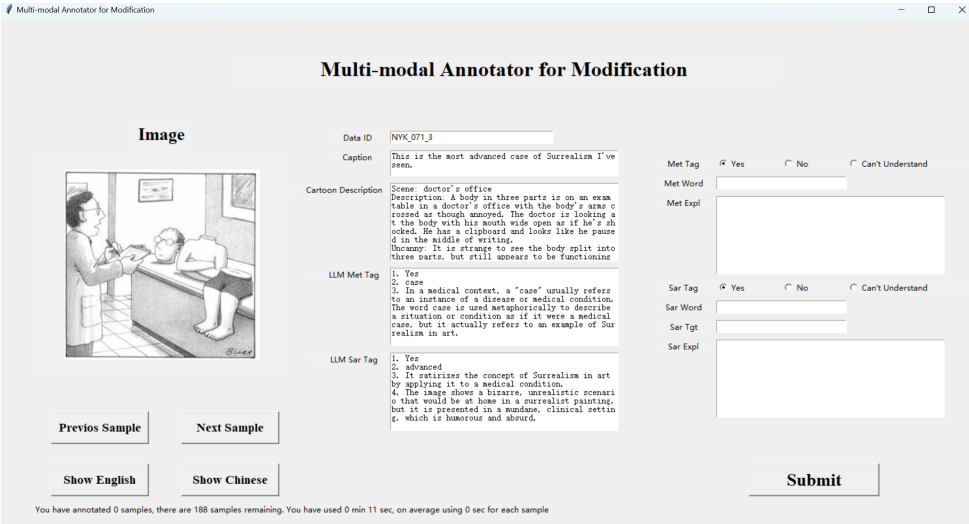


Figure 2: GUI for modifying step.

Round	Group	Pair-wise Kappa	000	001	011	111
Round 1	Group 1	0.0671, 0.0592, 0.2095	7	45	62	56
	Group 2	0.1164, 0.0608, 0.4068	25	62	46	37
	Group 3	0.1603, -0.0583, 0.0172	11	48	70	40
	Group 4	0.0730, 0.1732, 0.2929	12	29	63	66
	Total	-	55	184	241	199
Round 2	Group 1	0.3146, 0.2809, 0.4758	21	39	36	74
	Group 2	0.3360, 0.3744, 0.4826	37	45	32	56
	Group 3	0.4821, 0.4414, 0.5020	54	41	26	48
	Group 4	0.4226, 0.4226, 0.5713	28	32	29	81
	Total	-	140	157	123	259

Table 3: Consistency in classification step. The pair-wise kappa means the Kappa between annotator 1 and 2, 1 and 3, 2 and 3. The 000, 001, 011, 111 means the distribution of annotation results, for example, 001 means 1 annotator thinks it is positive, and another 2 annotators think it is negative.

- **GPT-4** (Achiam et al., 2023). GPT-4 is the best model in a lot of tasks, and support multi-modal input. We use GPT-4 as the text-only version.
- **GPT-4V**, the multi-modal version of GPT-4.

For GPT-3.5 and GPT-4, we input the description of cartoon instead of the image file. We run LLaVA-7B inference on an A40 GPU with 48G Video Memory, and run other models through Close-AI API. All of the experiments use the same prompt as 4, for text-only models, we delete the image file and use "Here is the description of the cartoon: (Insert the description)" instead. We use Macro P, R and F1 as evaluation metrics.

Table 8 shows the performance of large models on MC and SC tasks. We can find that all of these models can't do classification task well. For a model which can only output 1, the macro Recall is $(0\% + 100\%)/2 = 50\%$, but these large models' Recall on SC task is quite near to 50%, showing that their ability is quite low. Besides, we notice that all of them output 1 in most situation. We think that when we ask the models these questions, they will get a intuition that the input has deep meaning, so they will use their knowledge to make an explanation.

For MW, SW tasks, we use EM (Exact Match) and BLEU-4 as metrics; For ME, ST and SE tasks, since the answers are longer, we only use BLEU-4. Since the annotaion workflow takes GPT-4V's result as basic answer, we don't compare it with other 3 models. Table 9 and 10 shows the results. These results are consistent with the models' general ability and their size of parameters.

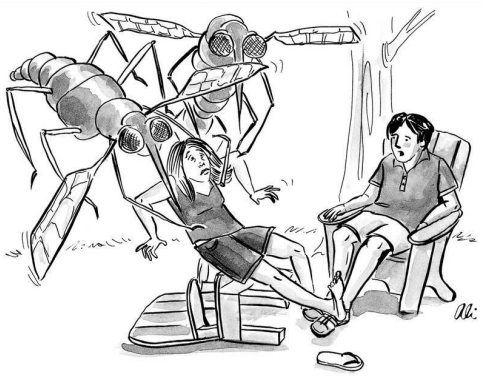
4.2 Fine-tuning Pre-trained Model Experiments

In this section, we designed a baseline model for MC and SC tasks, and improve its performance

Metaphor-Task Prompt	Sarcasm-Task Prompt
(Upload the cartoon file, such as 123.jpeg)	(Upload the cartoon file, such as 123.jpeg)
You are given a cartoon and its caption.	You are given a cartoon and its caption.
Caption: (Insert the caption)	Caption: (Insert the caption)
Please tell me:	Please tell me:
1. Does it contain metaphor?	1. Does it contain sarcasm?
2. If so, which word or object contains metaphor?	2. If so, which word or object contains sarcasm?
3. What is the word's real meaning?	3. What does it satirize?
4. Why do you think so?	4. Why do you think so?
You should answer these questions as brief as you can.	You should answer these questions as brief as you can.
For question 1, the answer must be Yes or No.	For question 1, the answer must be Yes or No.

Table 4: Prompt for GPT-4V annotation.

(a) A metaphor example. Caption: I smell a horse.



(b) A sarcasm example. Caption: Once they choose their queen, honey, it's really hard to change their minds.

Figure 3: Examples in NYK-MS.

Task	Pos.	Neg.	Total	Pos. Rate
Metaphor	928	214	1146	80.98%
Sarcasm	1142	2	1146	99.65%

Table 5: The classification result of GPT-4V. The sum of Pos. and Neg. is less than Total, because GPT-4V refused to annotate some samples, because the samples contain sensitive topics such as sex and politics.

Dataset	Pos.	Neg.	Total	Pos. Rate
NYK-M	1583	692	891	43.71%
NYK-S	1578	687	891	43.54%

Table 6: The scale of dataset NYK-MS.

Average MW Length	1.67
Average ME Length	24.22
Average SW Length	2.53
Average ST Length	6.90
Average SE Length	14.08

Table 7: The answer length of dataset NYK-MS.

with alignment and augment methods. The baseline model uses BERT (Devlin et al., 2019) as text encoder, and ViT (Dosovitskiy et al., 2020) as image encoder, and add a 2-layer MLP after them. We use multi-layer cross-attention to do cross-modal aggregate, and adding a classification layer to get the final output:

$$T = \mathrm{MLP}_T(\mathrm{BERT}(TextInput)) \quad (1)$$

$$I = \mathrm{MLP}_I(\mathrm{ViT}(ImageInput)) \quad (2)$$

$$A = \mathrm{CrossAttention}(T, I) \quad (3)$$

$$O = \mathrm{Softmax}(\mathrm{MLP}_C(A)) \quad (4)$$

Alignment

We use contrastive loss and Optimal Transport loss to do the alignment.

Before the Cross-Attention layer of baseline models, we add an alignment loss function. We experimented with two alignment loss functions: the contrastive learning loss function inspired by

Task	Model	Modality	Acc(%)	P(%)	R(%)	F1(%)
MC	LLaVA	Text+Image	47.59	54.37	52.19	42.70
	GPT-3.5	Text	47.85	58.88	53.09	40.95
	GPT-4	Text	43.78	**65.14**	51.37	33.19
	GPT-4V	Text+Image	**53.16**	64.93	**56.95**	**47.95**
SC	LLaVA	Text+Image	43.60	71.78	50.06	30.46
	GPT-3.5	Text	**46.32**	53.63	**51.42**	**39.87**
	GPT-4	Text	44.95	**72.41**	50.22	31.38
	GPT-4V	Text+Image	37.97	68.79	50.51	28.31

Table 8: The classification tasks experiments on zero-shot large models.

Task	Model	EM(%)	BLEU-4(%)
MW	LLaVA	29.75	33.60
	GPT-3.5	31.20	34.57
	GPT-4	**44.71**	**48.49**
SW	LLaVA	8.15	16.34
	GPT-3.5	18.34	25.73
	GPT-4	**32.65**	**42.19**

Table 9: The MW and SW tasks experiments on zero-shot large models.

Task	Model	BLEU-4(%)
ME	LLaVA	5.56
	GPT-3.5	6.57
	GPT-4	**10.11**
ST	LLaVA	0.95
	GPT-3.5	5.56
	GPT-4	**11.87**
SE	LLaVA	2.15
	GPT-3.5	3.46
	GPT-4	**5.28**

Table 10: The ME, ST and SE tasks experiments on zero-shot large models.

(Radford et al., 2021) and the OT Loss function inspired by (Villani et al., 2009). For comparison, the baseline refers to the case where no alignment is performed.

First, we introduced the commonly used contrastive learning method for modality alignment. Specifically, given the data in a batch $< T_1, I_1 >, < T_2, I_2 >, ..., < T_B, I_B >$, we calculate the contrastive loss from text to image and from image to text, and then take the average of these losses to obtain the final loss in a batch $Loss_{align}$.

$$sim(T_x, I_y) = \frac{T_x \cdot I_y}{||T_x|| \cdot ||I_y||}$$

$$L_{t\to i} = -\frac{1}{B}\sum_{x=1}^{B} \frac{e^{sim(T_x,I_x)/\tau}}{\sum_{y=1}^{B} e^{sim(T_x,I_y)/\tau}}$$

$$L_{i\to t} = -\frac{1}{B}\sum_{x=1}^{B} \frac{e^{sim(T_x,I_x)/\tau}}{\sum_{y=1}^{B} e^{sim(I_x,T_y)/\tau}}$$

$$Loss_{align} = \frac{L_{t\to i} + L_{i\to t}}{2}$$

Another method we use to align the text and images is the OT function. OT is a method to calculate the best transfer between two distributions with a cost matrix $Cost$. In the alignment task, we use the Euclid distance between embedding vectors as cost, and treat a batch as a uniform distribution of two modalities. So the task can be seen as a linear programming problem. We define that

$$p_i = \frac{1}{B}, i = 1, 2, ..., B \quad (5)$$

$$q_j = \frac{1}{B}, j = 1, 2, ..., B \quad (6)$$

$$Cost_{i,j} = dis(T_i, I_j) = ||T_i - I_j||_2^2 \quad (7)$$

and calculating Tr_{best} which satisfies

$$\sum_{j=1}^{B} Tr_{ij} = p_i, i = 1, 2, \cdots, B \quad (8)$$

$$\sum_{i=1}^{B} Tr_{ij} = q_j, j = 1, 2, \cdots, B \quad (9)$$

$$Tr_{ij} \in [0, \min(p_i, q_j)] \quad (10)$$

$$Tr_{best} = \text{argmin}_{Tr} \sum_{i=1}^{B}\sum_{j=1}^{B} Cost_{ij} Tr_{ij} \quad (11)$$

With $Cost$ and Tr_{best}, the loss function is

$$Loss_{OT} = \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} Cost_{i,j}(Tr_{best})_{ij}$$

Where B is the size of a batch. By using the total cost as the loss function and performing back-propagation on $Cost_{ij}$, the model can continuously optimize the distance between

Task	Model	Acc(%)	P(%)	R(%)	F1(%)
MC	Baseline(Text-only)	64.31	59.30	57.89	57.99
	Baseline(Image-only)	56.69	57.57	58.29	56.04
	Baseline	63.69	60.36	58.98	58.99
	Baseline + Contrastive Loss Alignment	63.06	59.16	59.00	59.07
	Baseline + OT Loss Alignment	64.33	61.31	61.65	61.44
	Baseline + VUA Data Augmentation	65.42	63.11	61.74	62.35
	Baseline + Knowledge Augmentation	**65.76**	**62.57**	**62.75**	**62.48**
SC	Baseline(Text-only)	59.87	59.56	59.36	59.34
	Baseline(Image-only)	57.32	57.30	57.34	57.26
	Baseline	60.60	60.63	60.18	60.02
	Baseline + Contrastive Loss Alignment	60.96	60.57	60.31	60.20
	Baseline + OT Loss Alignment	62.77	62.08	61.70	61.06
	Baseline + VUA Data Augmentation	**63.27**	**62.26**	**62.38**	**61.91**

Table 11: The classification tasks experiments on fine-tuning models.

vectors to achieve a more refined alignment effect. We use geomloss[4] library to solve the OT problem with Sinkhorn-Knopp algorithm (Knight, 2008).

Augmentation

First, we use data from VUA18 (Leong et al., 2018) and VUA20 (Leong et al., 2020) to do data augmentation on MC task. We select 1500 samples from them, adding these text-only samples with all-black images into train set. Then, we do augmentation on SC task similarly with HFM (Cai et al., 2019) dataset.

Finally, we use GPT-4 to generate the meaning and sample sentences of metaphor word, and give the baseline model these information to enhance its ability of recognizing the difference between literal meaning and in-context meaning. The detail of this knowledge method can be seen in Appendix E.

Experiment Details

Following our previous work (Zhang et al., 2024), we use Google's ViT-B_32[5] pre-trained model to obtain the image representation, and BERT-base-uncased from HuggingFace[6] to extract the text representation. The contrastive learning temperature coefficient τ is 0.1. The MLP layers after encoders adopt two-layer perception networks with a hidden dimension of 1536.

During the training, the batch size is set to 8, learning rate is 1e-5, dropout rate is 0.1. The model is trained for 15 epochs. The model employs a warmup strategy with a proportion of 0.1. The model parameter's L2 regularization coefficient is 0.01.

All fine-tuning experiments are conducted on a single NVIDIA 2080Ti GPU in less than one hour. For LLaVA experiments, we use an A40 GPU for inference in a few minutes.

Experiment Results

Table 11 shows the experiment of baseline model and these methods, where all of the results are the average value of 5 runs. We can see that single-modal input will decrease the results, and alignment and augmentation can improve the performance of models. In alignment method, the OT method is better than the contrastive method. The contrastive method may be not suitable for cartoons because these cartoons are too similar in general view.

5 Conclusion

In this paper, we describe our workflow for creating the new benchmark NYK-MS, and show the detail of it and examples. Our workflow can handle with difficult tasks where people can't get an agreement easily, and provide a base annotation for modification without any template or human-writing result. Based on this dataset, we conduct plenty of experiments. On large models, we show that large models can't do classification task well, and the ability on other tasks increases as the number of parameters get higher. On pre-trained base models, we use alignment and augmentation method to improve its performance, showing the effective and benefit for research of NYK-MS dataset.

[4] https://www.kernel-operations.io/geomloss/api/pytorch-api.html.

[5] https://storage.googleapis.com/vit_models/imagenet21k/ViT-B_32.npz.

[6] https://huggingface.co/bert-base-uncased.

Ethical considerations

There are totally 17 annotators, includin 2 undergraduate students, 11 MD students and 4 professional teachers. The gender ratio of annotators is 10(male):7(female).

We paid annotators 1 Yuan for each sample in each round. According to our statistics, the average speed of annotation is about 50 samples per hour, which means they can get 50 Yuan (6.9 dollars) per hour, far higher than the average salary in China.

Limitations

Our new dataset NYK-MS isn't a perfect dataset. First, it only contains image and text modalities, so it is useless for video, audio task; Second, the size of it is not very big; Third, the content in NYK-MS is cartoon and caption, so when we training models on it and do inference on other situations such as Tweets, the model's performance may be limited; Finally, even we have done lots of work, there may be some mistakes in NYK-MS. Furthermore, our opinion may be different with the authors of the cartoon and caption, and the debate on specific samples may still for long.

Besides, our alignment and augmentation methods also have limitations. These method can't understand the cartoon on object level, and can't build the relationship graph of objects and words. All of these method treat images as sequences as pixels, but for the cartoon, precisely locate the object is important.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arjun R Akula, Brendan Driscoll, Pradyumna Narayana, Soravit Changpinyo, Zhiwei Jia, Suyash Damle, Garima Pruthi, Sugato Basu, Leonidas Guibas, William T Freeman, et al. 2023. Metaclue: Towards comprehensive visual metaphors research. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23201–23211.

Muhammad Haseeb Aslam, Muhammad Osama Zeeshan, Soufiane Belharbi, Marco Pedersoli, Alessandro Koerich, Simon Bacon, and Eric Granger. 2024. Distilling privileged multimodal information for expression recognition using optimal transport. *arXiv preprint arXiv:2401.15489*.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European chapter of the association for computational linguistics*, pages 329–336.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in Twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, Florence, Italy. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Pragglejaz Group. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and symbol*, 22(1):1–39.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Jack Hessel, Ana Marasovic, Jena D. Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. 2023. Do androids laugh at electric sheep? humor "understanding" benchmarks from the new yorker caption contest. In *Proceedings of the*

61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 688–714, Toronto, Canada. Association for Computational Linguistics.

Lalit Jain, Kevin Jamieson, Robert Mankoff, Robert Nowak, and Scott Sievert. 2020. The New Yorker cartoon caption contest dataset.

Philip A Knight. 2008. The sinkhorn–knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275.

Luuk Lagerwerf and Anoe Meijers. 2008. Openness in metaphorical and straightforward advertisements: Appreciation effects. *Journal of Advertising*, 37(2):19–30.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 vua and toefl metaphor detection shared task. In *Proceedings of the second workshop on figurative language processing*, pages 18–29.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 vua metaphor detection shared task. In *Proceedings of the workshop on figurative language processing*, pages 56–66.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the fifth joint conference on lexical and computational semantics*, pages 23–33.

Teng Niu, Shiai Zhu, Lei Pang, and Abdulmotaleb El Saddik. 2016. Sentiment analysis on multi-view social data. In *MultiMedia Modeling: 22nd International Conference, MMM 2016, Miami, FL, USA, January 4-6, 2016, Proceedings, Part II 22*, pages 15–27. Springer.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Shraman Pramanick, Aniket Roy, and Vishal M Patel. 2022. Multimodal learning using optimal transport for sarcasm and humor detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3930–3940.

Dragomir Radev, Amanda Stent, Joel Tetreault, Aasish Pappu, Aikaterini Iliakopoulou, Agustin Chanfreau, Paloma de Juan, Jordi Vallmitjana, Alejandro Jaimes, Rahul Jha, and Robert Mankoff. 2016. Humor in collective discourse: Unsupervised funniness detection in the New Yorker cartoon caption contest. In *LREC*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. 2015. Inside jokes: Identifying humorous cartoon captions. In *KDD*.

Gerard J Steen, Aletta G Dorst, J Berenike Herrmann, Anna A Kaal, Tina Krennmayr, Tryntje Pasma, et al. 2010. *A method for linguistic metaphor identification*. John Benjamins Publishing Company Amsterdam.

Cédric Villani et al. 2009. *Optimal transport: old and new*, volume 338. Springer.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.

Yorick Wilks. 1978. Making preferences more active. *Artificial intelligence*, 11(3):197–223.

Bo Xu, Tingting Li, Junzhe Zheng, Mehdi Naseriparsa, Zhehuan Zhao, Hongfei Lin, and Feng Xia. 2022. Met-meme: A multimodal meme dataset rich in metaphors. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2887–2899.

Yingxue Xu and Hao Chen. 2023. Multimodal optimal transport-based co-attention transformer with global structure consistency for survival prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21241–21251.

Dongyu Zhang, Minghao Zhang, Heting Zhang, Liang Yang, and Hongfei Lin. 2021. MultiMET: A multimodal dataset for metaphor understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3214–3225, Online. Association for Computational Linguistics.

Ming Zhang, Ke Chang, and Yunfang Wu. 2024. Multi-modal semantic understanding with contrastive cross-modal feature alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11934–11943, Torino, Italia. ELRA and ICCL.

A Early Annotation Work

In our early work, we tried to use data from MVSA (Niu et al., 2016) and crawled from Twitter.

For MVSA dataset, we selected data with high inconsistency. MVSA is a sentiment anaysis dataset, and it gives each sample 2 or 6 sentiment scores. In MVSA-Single, 1 annotator gives the text and image a score s_t, s_i respectively; In MVSA-Multiple, 3 annotators give the text and image scores, so there is 6 scores. We use the average of them and finally calculated the text score s_t and image score s_i, then use $|s_t - s_i|$ as the metric of inconsistency. We sort the whole dataset and choose 3000 samples with highest inconsistency, and hire 3 annotators to check whether they contains metaphor. Since people use figurative language when they want to express some abnormal ideas, we believe that the inconsistency between modalities can lead to more metaphor samples.

For Twitter crawling, we use crawler and get some image-text pairs from Twitter, then do some post-process including removing advertisement, hiding personal information and deleting unrelated tags. We alse hire 3 annotators to annotate.

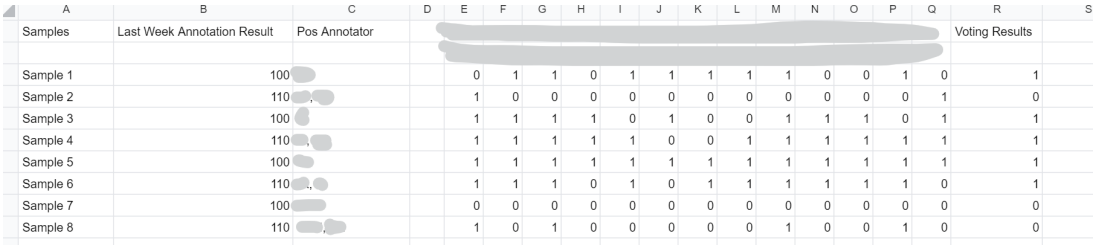
The final result is shown in Table 1. We found that the positive rate is too low, and most of the positive cases are too easy to recognize even without the image. So, we give up this method and then choose cartoon-caption pairs as our source data.

B GUI Details

A Graphical User Interface that shows all information on the screen can help a lot when annotating. This process is not only tedious but also prone to inconsistencies and errors. Therefore, we develop a visual annotation program that implements data display, assisted reading, automatic recording, and time statistics functions. This significantly improves annotation efficiency and lays a solid foundation for other tasks and future work.

Basic functions

This program is implemented using the tkinter visualization library in Python, achieving basic functions: Upon program startup, automatically search for unannotated files in the current directory and load the first unannotated data entry. When loading data, display the image, title, and image description from the original dataset in specified locations. Provide annotation fields for each task on the interface, using radio buttons for tasks that



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	Samples	Last Week Annotation Result	Pos Annotator															Voting Results	
	Sample 1	100			0	1	1	0	1	1	1	1	1	0	0	1	0	1	
	Sample 2	110			1	0	0	0	0	0	0	0	0	0	0	0	1	0	
	Sample 3	100			1	1	1	1	0	1	0	0	1	1	1	0	1	1	
	Sample 4	110			1	1	1	1	1	0	0	1	1	1	1	1	1	1	
	Sample 5	100			1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	Sample 6	110			1	1	1	0	1	0	1	1	1	1	1	1	0	1	
	Sample 7	100			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Sample 8	110			1	0	1	0	0	0	0	0	1	0	0	1	0	0	

Figure 4: Discuss screenshot.

determine the presence of metaphors or sarcasm, and text boxes for other tasks.

Auxiliary functions

According to the annotator's suggestions, we provide reference suggestions from ChatGPT and translation functions via API. The ChatGPT's suggestions are pre-called and stored offline, while the translation function calls the API in real-time.

During annotation, we found that many comics contain elements related to religion and foreign politics, which are very unfamiliar to the annotators. Therefore, additional knowledge assistance is required for proper understanding. Our approach is to use the output of ChatGPT as a reference, with the final annotation still performed by humans. Specifically, before annotation, we will provide ChatGPT with the prompt in Table 12:

Explanation Prompt
I will give you the description of a cartoon and its caption. Please tell me whether the caption and cartoon contain metaphor(sarcasm), and explain your thought in detail. Description: (description for the data) Caption: (caption of the data)

Table 12: Prompt for Explanation.

We use Baidu's translation API to provide real-time translation functionality. When the annotator clicks the corresponding button, the program will upload the comic description and the explanations generated by ChatGPT to Baidu's translation platform via the API, then retrieve the translated Chinese content, and display it on the interface.

C Annotation Rules for Human

Here is some of our annotation rules:

- If there are multiple metaphor/sarcasm words, choose the most obvious one;
- Pun and homophonic are not metaphor; (For example, "killer shark")
- simple or common combinations are not metaphor; (For example, "give up")
- Make sure that the sarcasm can show the speaker's negative sentiment, otherwise it is just normal humor;
- If possible, the MW, SW, ST answers should be words in the caption or cartoon description. Otherwise, a special token must be add.
  - Word from caption: No token
  - Objects in image: [I] + Object (Must make sure that the name of object is in the description text)
  - The whole image: [I] (Only for metaphor word and sarcasm word annotation)
  - Otherwise: [S] + Content (Free-writing, as brief as you can)

Besides, we selected some examples as references, and teaching the annotators to follow these cases.

D Discussion Details

Figure 4 is one of our discussion screenshots. In this discussion, we show 8 examples which are annotated as positive samples by only 1 or 2 annotators. We invite them to explain their idea, and then vote to decide the final annotation result. These samples then will be listed into annotation rules as reference.

E Knowledge Augmentation Method

We selected the samples in NYK-M training set which satisfies that the sample is positive sample, and the MW annotation result is a substring of the caption.For these samples, we use the MW annotation result as target word; for other samples, we

Word Meaning Prompt
1. What's the basic meaning of word "(target word)"? 2. Please write a sample sentence to demonstrate this meaning. You should answer these questions as brief as you can.

Table 13: Prompt for Explanation.

selected the longest word as target word. Then, we use the prompt in Table 13 for asking GPT-4.

We use the answer of question 2 as the sample sentence of basic meaning. When using BERT to get the embedding of captions T_C, we additionally calculate the embedding vector of the sample sentence T_S. Then, we get the token level embedding for the target word (average for multi tokens) $T_{C,i}$ and $T_{S,j}$, where i, j is the index of target words. We want the distance between basic meaning and in-context meaning farther for metaphor words, so we use such loss function: $Loss_{knowledge} = y \times sim(T_{C,i}, T_{S,j}$ Where sim is cosine similarity and y is -1 for samples using metaphor words as target words, for other cases (including negative samples and MW not in caption), y is 1.

Table 14 shows an example where we want to make the distance of embedding vectors of word "flat" in caption and sample sentence farther.

Data ID	NYK_722_3
MC	1
MW	**flat**
Caption	The piano's in tune, but the house is a little **flat**.
Meaning	The basic meaning of "**flat**" is having a smooth, even surface without any bumps or angles.
Sample	The lake was so calm that the water looked completely **flat**.

Table 14: Meaning Augmentation result.