Fine-tuning and Sampling Strategies for Multimodal Role Labeling of Entities under Class Imbalance

Syrielle Montariol[†*] and **Étienne Simon**[‡*] and **Arij Riabi**[†] and **Djamé Seddah**[†]

[†]INRIA Paris
F-75012 Paris, France
firstname.lastname@inria.fr

[‡]Sorbonne Université, CNRS, ISIR
F-75005 Paris, France
etienne.simon@isir.upmc.fr

Abstract

We propose our solution to the multimodal semantic role labeling task from the CONSTRAINT'22 workshop. The task aims at classifying entities in memes into classes such as "hero" and "villain". We use several pre-trained multi-modal models to jointly encode the text and image of the memes, and implement three systems to classify the role of the entities. We propose dynamic sampling strategies to tackle the issue of class imbalance. Finally, we perform qualitative analysis on the representations of the entities.

1 Introduction

Social media memes can be defined as "*pieces of culture, typically jokes, which gain influence through online transmission*" (Davison, 2012). More specifically, memes are visual templates usually associated with a textual caption. Analysing memes involves many unique challenges that differ from classical multimodal tasks such as image captioning and visual question answering. While unimodal models can often perform well on multimodal datasets (Agrawal et al., 2018), memes involve a lot of entanglement – stylistic or semantic – between the two modalities, such as the caption contradicting the image. This makes memes intrinsically multimodal. Furthermore, pragmatics – the context's contribution to meaning – plays a key role in the interpretation of memes. In particular, phenomenons such as irony are challenging to detect. Even human annotators have difficulties in interpreting a meme correctly without knowledge of the community in which the meme was shared.

In this paper, we tackle the shared task on multimodal semantic role labeling of the CONSTRAINT'22 workshop (Sharma et al., 2022). Given a (meme, entity) pair,[1] the goal is to classify the entity's role in the meme into one of four classes (`hero`, `villain`, `victim` or `other`) from the perspective of the author of the meme. The multimodality of the problem stems from the meme, which is given as an (image, OCR) pair, where OCR (for Optical Character Recognition) is the caption extracted from the image. The dataset covers one language, English, and two domains, COVID-19 and US politics. Figure 1 shows a sample from the training set.

Understanding memes involves a lot of commonsense and cultural knowledge on the political stance of the entities. Thus, it requires models pre-trained on a large amount of data, capable of recognising key entities such as political figures in both modalities, and of inferring their relationship, their role and the public opinion of a community on them. To evaluate the task's difficulty, we manually annotate a set of samples. With 5 annotators, we reach an average Macro-F_1 of 0.65 (see details in Appendix A), less than 10 points above the best system submitted to the shared task.

We propose systems relying on several multimodal (vision–language) pre-trained models: One For All (OFA, Wang et al., 2022), CLIP (Radford et al., 2021) and VisualBERT (Li et al., 2019). We use these models as encoders to extract multimodal meme representations. These *encoders* are introduced in Section 3. We then design several neural network classifiers to handle these representations in a task-specific fashion. These *classifiers* are presented in Section 4.1.

The CONSTRAINT'22 dataset is characterised by a large class imbalance, with the most frequent class gathering 78% of the samples in the train set, while the least frequent one is conveyed by less than 3% of the samples. However, the challenge is evaluated using a Macro-F_1 metric and calls for balanced performances across all classes. To handle this discrepancy, we developed several sub-

[*]These authors contributed equally.

[1]We take each (meme, entity) pair as independent samples, thus considering all entities of a meme independently during training and inference.

Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations, pages 55 - 65

Figure 1: In this meme, the OCR is: "WEARS A MASK THE SAME WAY\nEXIT\nHE HANDLES THE\nPANDEMIC \nmakeameme.org\n". There are two entities, "Donald trump" labeled as `villain` and "mask" labeled as `other`.

sampling strategies that we present in Section 4.2.

Our best results are obtained by ensembling predictions from all of our models, using various ensembling methods. The details of the ensembling methods are given in Section 4.3. Finally, we present our performance in Section 5 along with a qualitative analysis of our models. We highlight the limitations of the dataset, task and methods in Section 6.

To summarise, our whole architecture is built on freely available pre-trained models. We only fine-tune these models for the multimodal semantic role labeling task. This makes computational training cost particularly low. Our system can be characterised by:

- Simple classifier design on top of deep pre-trained model.
- Handling of class imbalance through carefully-designed sampling strategies.

Our code is available at: `https://github.com/smontariol/mmsrl_constraint`.

2 Related Work

Multimodal semantic role detection in memes is a relatively unique task, compared to other language–image multimodal task such as object classification and entity action detection, it requires a lot more contextual and cultural background. In this section, we list some related problems before introducing tools to tackle the task at hand in the next section.

In recent years, social media platforms have seen a wave of multimodal data in diverse media types. This attracted the interest of researchers to combine modalities to solve various tasks with joint representations, where the model's encoder takes all the modalities as input, or separated representations, where all modalities are encoded separately (Baltrušaitis et al., 2018).

In the CONSTRAINT'22 challenge, we tackle multimodal semantic role labeling (SRL). SRL is originally a Natural Language Processing (NLP) task which consists in labeling words in a sentence with different semantics roles to determine Who did What to Whom, When and Where (Gildea and Jurafsky, 2002; Carreras and Màrquez, 2005); these roles are also known as thematic relations. It was extended to the computer vision domain through Visual SRL. Visual SRL benchmarks focus on situation recognition in images (Silberer and Pinkal, 2018; Pratt et al., 2020); these tasks heavily rely on object detection systems for visual groundings (Yang et al., 2019). This differs from the methods we need to implement for the shared task, where the entities do not necessarily appear in the image. Moreover, in our case, the semantic role is taken from the point of view of a political argumentative: the perception of the entity by the author of the meme. This involves completely different features compared to labeling the thematic relations of the entity; in particular, cultural and contextual knowledge on the background of the meme.

Another similar task is multimodal named entity recognition, which aims at identifying and classifying named entities in texts and images. It requires more in-domain knowledge compared to multimodal SRL; but most multimodal NER datasets are text-centric, with the image being an additional feature for the text-based prediction (Arshad et al., 2019; Chen et al., 2021), while our task is more symmetrical or even image-centric.

Finally, many shared task on memes have been proposed in recent years, with a large variety of tasks: emotion classification (e.g. MEMOTION task at SemEval 2020 Sharma et al., 2020); hateful meme detection (e.g. the Hateful Meme Challenge Kiela et al., 2020) event clustering (e.g. DANKMEMES at EVALITA 2020 (Miliani et al., 2020)); more fine-grained hateful content analysis (Fine-Grained Hateful Memes Detection Mathias et al., 2021, aiming at classifying the target attacked by the meme and the type of attack); or and detection of persuasion techniques (e.g. Semeval 2021 Task 6, Dimitrov et al., 2021).

3 Multimodal Encoding

Since we experiment with deep neural networks, we need to obtain distributed representations of our inputs. To this end, we use pre-trained mod-

els with good performances on popular datasets. These models are multimodal transformers, that we use to encode image and caption's OCR into a common latent space. While transformers were originally developed for natural language processing (Vaswani et al., 2017; Devlin et al., 2019), they subsequently became ubiquitous in computer vision models as well (Dosovitskiy et al., 2021). To process an image, it is first cut into a sequence of $P \times P \times C$ patches. These patches are then projected into the transformer input dimension, either using a single linear layer, or using a full-fledged CNN architecture.

The output of a transformer has the same length as its input. We call this length N; it is the number of patches in the image, the number of tokens in the OCR, or the sum of the two for multimodal transformers. Thereafter, we refer to an encoded meme image i and OCR o as $\mathrm{enc}_{\text{full}}(o, i) \in \mathbb{R}^{N \times d}$. This output can be further pooled into a fixed-size representation $\mathrm{enc}_{\text{pool}}(o, i) \in \mathbb{R}^{d}$. We now describe what models are behind these encoder functions.

3.1 CLIP and VisualBERT

The multi-modal features are extracted from the caption's OCR and the meme image using two vision-language models, CLIP and VisualBERT.

CLIP (Contrastive Language–Image Pre-training, Radford et al., 2021) is trained using text as supervision to encode images, with 400 million image–text pairs available on the internet. The training task is to predict which text is associated with an image, from all text snippets of the batch, using a contrastive objective instead of a predictive one for computational efficiency. CLIP trains an image encoder and a text encoder jointly, maximizing the cosine similarity of the image and text embeddings in the joint representation space for positive pairs, and minimizing similarity of negative pairs. The strength of this task is to offer large robustness and zero-shot capability to the model, to transfer to many classification tasks. Image encoding is done using a variation of the Vision Transformer (ViT, Dosovitskiy et al., 2021). Text encoding is done using a GPT-like language model (Radford et al., 2019).[2]

Similar to CLIP, we use a VisualBERT model (Li et al., 2019) trained on visual commonsense reasoning and image captioning. VisualBERT uses self-attention to align parts of the text with regions of the image and build a joint representation. It mostly differs from CLIP in its training procedure in three phases: task-agnostic pre-training, task-specific pre-training, and task-specific fine-tuning. Moreover, VisualBERT does not include an image encoder; the patch features are extracted beforehand with pre-trained image classification and segmentation models. We extract features using FasterRCNN (Ren et al., 2015), EfficientNet (Tan and Le, 2019) and VGG (Simonyan and Zisserman, 2015). Bucur et al. (2022) showed that EfficientNet features prove useful for sentiment and emotion analyses of meme, while Pramanick et al. (2021) prove the efficiency of VGG for detecting harmful memes and identifying their target.

The output of both CLIP and VisualBERT can either be pooled ($\mathrm{enc}_{\text{pool}}$) or be used as-is ($\mathrm{enc}_{\text{full}}$).

3.2 OFA

A second method we experiment with to obtain a distributed representation of text and images is OFA (One For All, Wang et al., 2022). OFA is based on an encoder–decoder architecture pre-trained on several visual, textual and cross-modal tasks. A key point of OFA is to leverage a diverse set of training tasks to obtain good zero-shot performances. Despite this claim, we did not obtain satisfactory zero-shot results. We hypothesize that this is due to the noisy OCR and to the nature of meme role labeling which is radically different from what OFA was pre-trained on.

All tasks are expressed as sequence-to-sequence problems, such that a single OFA model can be used without the need of task-specific layers. For example, one of the pretraining task is image captioning; for this task, the model is trained to predict the caption given the image and the text "What does the image describe?" as inputs.

The input image and text are fed jointly to the encoding transformer using modality-specific positional embeddings. The image representation is built from 16×16 patches embedded by a ResNet (He et al., 2016). The decoding transformer is trained as a causal language model conditioned on the encoder's output with a standard cross-entropy loss. When the output is constrained on a small number of classes, the model is trained and evaluated on the task's output domain, not on the whole output vocabulary.

[2]The sequence length is limited to 76 byte-pairs. In the CONSTRAINT task corpus, 76 byte-pairs corresponds to the 95th quantile of OCR text length in the test set, and slightly more in the train set.

For the meme role labeling task, we feed OFA with the image as well as the following instruction:

"What is the category of ENTITY *between hero, villain and victim?* OCR*"*

As we detail in the next Section 4, we train OFA either as a sequence to sequence problem (resulting in a pair of models enc_{OFA}–dec_{OFA}) or by adding a classification head on top of the decoder (which can be used as a standard enc_{pool}).[3]

4 Models

We now describe how we use the encoded text and images for semantic role labeling.

4.1 Classification

We experiment with three different methods to classify a $(\text{meme}, \text{entity})$ pair, depending on what kind of representation we get from the encoder. The representation of the meme is composed of the image's representation along with the encoded caption's OCR, and any extra features such as the list of entities related to the meme. For ease of notation, we group under "OCR" all extra features which were extracted from the meme, and we refer to them using a single variable $o = (\text{OCR}, \text{caption}, \dots)$. Image features are referred to by i and the encoded list of entities by e. All classifiers are illustrated in Figure 2.

Multilayer perceptron (MLP) When the output of the encoder is of fixed size, we use a 2-layers MLP classifier. The input of the classifier is made from the encoding of the OCR, image and entity. The representation of the entity is obtained using the same transformer used to process the OCR. The output of the model is a softmax on the four possible roles:

$$P(r \mid o, i, e) \propto \exp \text{MLP}\left(\begin{bmatrix}\text{enc}_{\text{pool}}(o, i) \\ \text{enc}_{\text{pool}}(e)\end{bmatrix}\right)_r.$$

This model is trained using a standard cross-entropy loss. Depending on the encoder, we either train the MLP alone, or the MLP and the encoder jointly.

Attention When the representations of the OCR and image are not pooled along the sequence's length, we use an attention mechanism. In this case, the query of the attention is the entity we wish to classify, while the memory is built from a concatenation of the image and OCR encoded by CLIP or VisualBERT:

$$\alpha_j \propto \exp\left(\text{enc}_{\text{pool}}(e)^\intercal \boldsymbol{W}_k \, \text{enc}_{\text{full}}(o, i)_j\right),$$

$$\boldsymbol{a} = \text{ReLU}\left(\sum_j \alpha_j \boldsymbol{W}_v \, \text{enc}_{\text{full}}(o, i)_j\right),$$

where $\boldsymbol{W}_k$ and $\boldsymbol{W}_v$ are parameters used to project the encoded meme for use as attention key and value. We classify the attention output $\boldsymbol{a}$, using a softmax layer $P(r \mid o, i, e) \propto \exp(\boldsymbol{W}_p \boldsymbol{a})_r$.

Since the encoders already use positional embeddings, we do not add this information to our classifier's attention. However, we do use segment embeddings to distinguish the vectors encoding the image, OCR or entity list in the encoder's output. We use different MLP layers depending on whether a vector correspond to an input image, OCR or entity list. This model is also trained by minimizing the cross-entropy with gold labels.

Seq2seq When using an OFA encoder, we also attempt to stay in the sequence to sequence framework and train the model to generate the class labels. In this case, if we denote the label's tokens by $\boldsymbol{\ell}$, the model is trained to maximize the likelihood that the meme (o, i) has the gold target $\boldsymbol{\ell}$:

$$P(\ell_k \mid \boldsymbol{\ell}_{<k}, o, i) \propto \text{dec}_{\text{ofa}}(\text{enc}_{\text{ofa}}(o, i), \boldsymbol{\ell}_{<k})_{\ell_k},$$

where $\boldsymbol{\ell}_{<k} = [\ell_1, \ell_2, \dots, \ell_{k-1}]^\intercal$ refers to the list of previous tokens. To evaluate this model, the log-likelihood of the possible labels are summed along sequence length:

$$\hat{r} = \arg\max_r P(r \mid o, i) \propto \prod_k P(\ell_k^{(r)} \mid \ell_{<k}^{(r)}, o, i),$$

where $\boldsymbol{\ell}^{(r)}$ designates the list of tokens for the label r, such as $[\texttt{vil}, \texttt{lain}]^\intercal$.

Additional features As explained in Section 2, our task is quite different from most multimodal tasks on which the encoders were trained; it is much more abstract and requires a lot of additional background knowledge. Thus, when using CLIP and VisualBERT, we add supplementary features as input to the classification model (MLP and attention).

We add as textual features the list of entities associated with the meme, this list is directly available in the dataset. We encode the entities' names

[3] For the OFA model, enc_{pool} refers to the output of the penultimate layer of OFA's decoder, while we use enc_{OFA} to reference only the OFA's encoder.

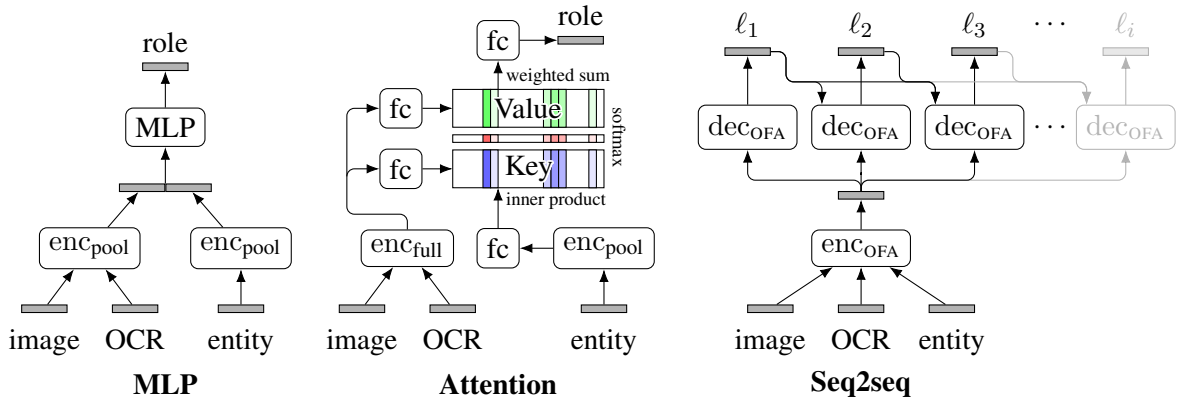

Figure 2: Our three classifiers. Note that each classifier uses a different combination of encoders. MLP is used with enc_{pool}, Attention requires enc_{full}, while Seq2seq requires an enc_{OFA}–dec_{OFA} pair.

using the same encoder as the system (CLIP or VisualBERT).[4] We also add to the system the image features that were extracted using VGG, EfficientNET and FRCNN.

4.2 Dealing with Class Imbalance

The dataset faces a large class imbalance, with the class `other` being over-represented (78% in the train set) and classes `hero` and `victim` consisting of only 2.7% and 5.2% of the train set respectively. Thus, training on the raw dataset might lead to overfitting and over-predicting the majority class. Moreover, recall that the evaluation metric is Macro-F_1, which weighs each class equally; hence the importance of solving the class imbalance issue.

Our first solution was to weight labels in the loss. This loss penalisation led to poor performances; we suspect this is due to the working of the optimization algorithm we used. Adam and its variants estimate the distribution of the gradients using exponential moving averages; these estimates are faulty when the magnitude of the loss changes often.

A common strategy is over-sampling the low-frequency classes and under-sampling the high-frequency ones. Each (meme, entity) pair is dropped with a pre-defined probability, following various class sampling strategies. We evaluated 6 different sampling strategies illustrated in Figure 3:

Micro does not subsample. This optimize the Micro-F_1, which puts more weight on samples labeled `other` due to their sheer number.

Macro subsamples memes such that the label distribution is uniform. This implies dropping a large amount of `other` samples in order to lower their frequency.

In-between is a compromise between *micro* and *macro*, balancing between matching the evaluation loss and seeing a more diverse set of samples.

Interpolate drifts from *micro* to *macro* during training. For the first epoch, the memes are sampled according to the empirical distribution (*micro*); while the last epoch is sampled to have a uniform label distribution (*macro*).

Cycle alternates between *micro* and *macro* (2-epoch *short cycle*) or between *micro*, *macro* and two different *in-between* (4-epoch *long cycle*).

For the last two strategies, the sampling rates are updated at the end of each epoch during training. In general, these *dynamic* sampling strategies performed better than sampling strategies with a fixed rate for the whole training duration.

4.3 Ensembling

In order to further improve our results, we build several ensemble of our models. We filter-out models with a low validation macro-F_1 and experiment with several ensembling techniques. Due to the small size of the dataset, we did not create an additional split to evaluate our ensembling approach. In

[4] We also experiment with adding generated captions as features. We generate them using an OFA model trained for automatic caption generation. However, the captions are very generic and descriptive; for example the entities names are not captured by the model. This features does not improve the systems, hence we do not further develop it in the results section.

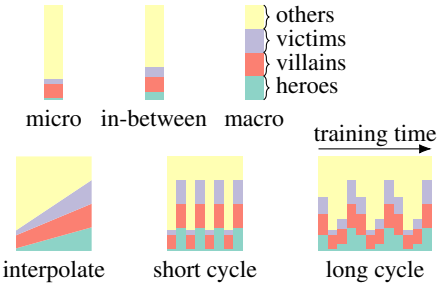

Figure 3: Target frequencies of the various strategies during training. The *micro* strategy corresponds to using the empirical class distribution in the dataset, that is hero 2.7%, villain 13.9%, victim 5.2% and other 78.2%.

this context, overfitting the validation set is a risk. Two of the ensembling methods we evaluate are therefore non-parametric. These non-parametric strategies take the average or the median probability assigned to each class by all models.

Preliminary results indicate that training a linear model to weight the output of our various models is tedious and does not improve over non-parametric strategies. We therefore turn towards gradient boosted trees (Friedman, 2001) trained by XGBoost (Chen and Guestrin, 2016). XGBoost builds an ensemble of decision trees, whose internal nodes correspond to conditions on our models' output, and whose leaves correspond to a predicted semantic role. Boosted trees have the potential to outperform non-parametric methods by better capturing the scale of various models' output, however it has the downside of being very prone to overfitting.

5 Results

5.1 Experimental process

The train set consists of 17 514 (meme, entity) pairs, the validation set 2 069 pairs and the test set 2 433 pairs. We did all the training on the datasets from the two domains, COVID-19 and US politics jointly. The test set contains examples from both domains. The evaluation is done with Macro-F_1 score; the OCR and the list of entities are provided along with the image of the meme. We run all experiments 5 times to check for the robustness of results and perform statistical testing.

For CLIP, we use the biggest `L/14` CLIP-ViT model built on the Vision Transformers (Dosovitskiy et al., 2021). Both preliminary self-supervised fine-tuning and fine-tuning while doing the classification failed. This is probably due to the size and the format of the shared task dataset, much smaller and quite different from the training data of the pre-trained model; any fine-tuning leads the model to forget the knowledge it learned during pre-training. Consequently, we freeze all layers and tune only the classifier, with the architectures described in Section 4.

For VisualBERT, we fine-tune the `visualbert-vcr-coco-pre` model trained on caption generation and visual commonsense reasoning.

For OFA enc_{pool} with an MLP classifier, we obtained better results by fine-tuning the whole model from the `vqa_large_best` checkpoint[5] using a small 0.1 label smoothing and feeding the OCR and entity both to the encoder – along with the image – and to the decoder. Our OFA seq2seq model follows the same setup using the `ofa_base` checkpoint.

In the dataset, several entities are associated with more that one label. As this situation is infrequent, we consider the small amount of samples with multiple labels does not warrant a full-fledged multi-label classification setup. Thus, our models output a single categorical distribution. When multiple labels ought to be predicted for an entity (the entity appears twice in the list of entities associated with the meme), we predict them in order of likelihood.

5.2 Quantitative results

Classifier results. Table 1 compares our main models on the CONSTRAINT'22 test set. We measure the statistical significance of our results using a one-sided Welch's unequal variances t-test (Welch, 1947) under the null hypothesis that the macro-F_1 are equals. Some hyperparameters are optimized on a per-model basis. In particular, using the list of entities as additional feature improves the performance for VisualBERT and CLIP-attention but not for our best CLIP-MLP model.

A CLIP enc_{pool} together with an MLP classifier reached the best performances among our non-ensembling model pool, significantly ($p < 0.0004$) improving over the OFA MLP combination. Using the unpooled features of the transformers (enc_{full}) with an attention classifier underperform compared to the enc_{pool}+MLP approach. However this difference is not significant in the case of VisualBERT ($p < 0.3$). In particular, attention-based

[5]This refers to an OFA model pre-trained on 8 tasks then fine-tuned on VQA from the official OFA repository.

Encoder	Classifier	Macro-F_1	
		mean	std
OFA	MLP	44.6	0.5
OFA	Seq2seq	44.0	0.9
CLIP	MLP	**47.0**	0.5
CLIP*	Attention	42.3	1.7
VisualBERT*	MLP	43.1	0.2
VisualBERT*	Attention	42.3	1.8
Ensemble mean		47.9	-
Ensemble median		47.5	-
Ensemble XGBoost		47.6	-
Challenge's top score		58.7	-
Human		65.5	4.6

Table 1: Comparison of the best systems with the different encoders and classification architectures. All systems are run 5 times with 25 epochs. Encoders with a * in exponent are augmented with the list of entities as feature.

Sampling	Macro-F_1	
	mean	std
micro	38.3	1.0
in-between	44.1	0.3
macro	42.3	0.6
interpolate	46.3	0.8
short cycle	**47.0**	0.5
long cycle	46.5	0.5

Table 2: Sampling results with the CLIP model and MLP classifier, with 500 batch per epoch.

approaches have more variance than their MLP counterpart. The OFA seq2seq model reaches performances within the error margin of the OFA MLP model ($p < 0.14$), which is not surprising since the two models are relatively close. The gap between VisualBERT and OFA is somewhat significant with p-values between 0.001 and 0.07 depending on the pairwise comparison. As expected, ensembling leads to the best result, regardless of the ensembling strategy; human annotators far exceed current model performances. We further develop human annotation in Section 6.

Sampling results. Table 2 compares the different sampling strategies represented in Figure 3 for training a CLIP encoder with MLP model. As expected, using the empirical class distribution (*micro* strategy) leads to the worse score. While the *macro* strategy is in theory what we should maximise to improve the Macro-F_1, it is second worst among all strategies. The dynamic strategies, which use evolving sampling frequencies during training clearly outperform static strategies. In particular, for training CLIP, the *short cycle* strategy outperforms the other ones, but the difference with *long cycle* and *interpolate* is not statistically significant (p-values > 0.05). We observe similar tendencies with systems based on OFA and VisualBERT, with a slight advantage to the *interpolate* strategy over the *cycling* ones for the former.

Despite the different subsampling strategies, the per-class performances vary widely, see for example the results for the CLIP MLP model with a *short cycling* subsampling strategy:

%	hero	villain	victim	other
F_1	20	50	33	84
Precision	15	46	26	90
Recall	33	56	45	79

We observe similar results with all hyperparameter combination. These performances somewhat follow the empirical distribution of the classes, with the rarest class `hero` having the worst performance, and `victim` being not much better. This makes us consider sub-sampling `other` even below 25%. However, this observation-inspired "*super-macro*" strategy did not prove successful, reaching an average Macro-F_1 or 40.0, higher than the *micro* strategy but lower than the *macro* one.

5.3 Qualitative analysis

We extract the embeddings of all entities in the train set as their are embedded by the CLIP model, right before being fed into the MLP or being used as query for the attention mechanism. Keeping only the ones occurring more than 30 times, we perform a PCA on their embeddings and represent the first two components in Figure 4. Each point represents an entity, its colour depends on the distribution of labels that are attributed to the entity, normalised by the global frequency of each label in the full dataset. We keep only the two most frequent labels associated with the entity for colouring. We can see that inanimate objects tend to be labeled as `other`. On the other hand, large political parties are nearly always portrayed as `villain` with America as a `victim`. The somewhat unexpected heroic status of the libertarian party can be explained by the pres-

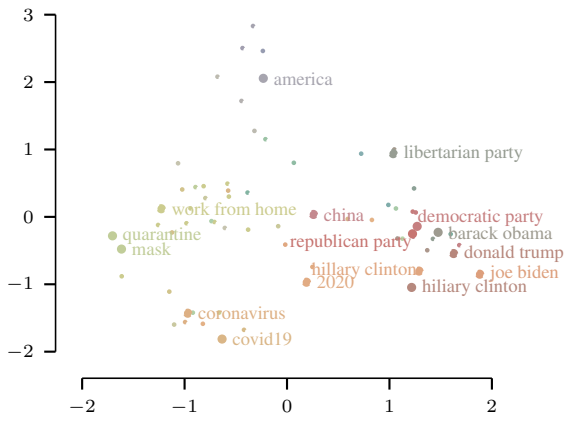

Figure 4: PCA of entity embeddings from CLIP. The explained variance is 33%+18%. The entities appearing more than 30 times, with labels attached to the 16 most frequent ones. The color of the embeddings reflect the role attached to the entity in the train set (`hero`, `villain`, `victim`, `other`). When the entity is assigned different roles, the color are mixed together; e.g. covid19 appears twice as often as `other` as it does as `villain`.

ence of advertisements in the form of memes in the dataset. We can see that CLIP was able to separate the entities according to their probable class even before processing the meme. Still, the model can't clearly distinguish between most heroes and villains without seeing the meme, which is to be expected.

6 Discussion

The multimodal aspect is crucial in this task. When looking at entity names, only 15% have an exact surface form match in the caption's OCR; moreover, the OCR is often incomplete or noisy (see example in Figure 1 with the "Exit" sign popping in the middle of the caption). Thus, using only the text is far from sufficient. On the other hand, recognising the entities in the image of the meme is not an easy task. As stated in the introduction, the image and the text are often not directly related. Moreover, the image often contains elements not seen in common image datasets; for example, meme creators often perform montages like swapping faces and objects. Overall, a lot of commonsense and cultural knowledge is needed for the model to understand what the meme is about.

The absence of contextual information also makes the task difficult for humans. To evaluate the difficulty of the task, we performed human annotation of a sample of 100 (image, entity) pairs with five annotators. Details of annotation process can be found in Appendix A. The average pairwise Cohen's κ (Cohen, 1960), used to measure the inter-annotator agreement, is 0.47. It indicates a "moderate" agreement according to Cohen (1960). However, it also shows that less than one third of the annotations are reliable (McHugh, 2012). Moreover, the macro-F_1 scores are relatively low: the average is 0.65 and the maximum 0.69. Having metadata such as source website and date of publication of the meme would help human and algorithmic annotators alike.

Finally, from a real-world point of view, this task is not entirely complete: the OCR and the list of entities are already provided in the dataset, and we only have to perform the classification. In a real-life setting, we would create a multi-task system jointly extracting the caption, detecting entities and classifying them; the three tasks complementing each other.

7 Conclusion

In this work, we propose several systems to solve the task of classifying entity roles in memes. We focus on comparing classification models – MLP, Attention and Seq2seq systems – on top of pre-trained multimodal encoder: CLIP, VisualBERT and OFA. Our best standalone system uses the CLIP encoder with MLP classifier, but our best score is obtained using ensembling of a large number of models. We also compare several sampling strategies to deal with the class imbalance issue, proposing dynamic sampling methods that outperform the standard uniform ("*macro*") sampling.

As a preliminary future work, more or less straightforward processing can be performed on the dataset, at the entity-level (using an entity linker to resolve surface forms to entity identifiers, e.g. merging entities "US" and "United States" together); at the OCR-level (performing lexical normalization (Samuel and Straka, 2021) to deal with OCR errors and meme-specific syntax); and at the image-level (removing the text from the image, for a less noisy image embedding).

To improve the model, entity representation is key. We wish to train global entity embedding, shared across the whole dataset, and contextualised entity embeddings, aligning the entity's vector representation in the image and in the OCR of the meme (when there is an explicit mention of it).

8 Acknowledgments

We want to express our strong gratitude to Matt Post for the time he took providing manual annotation for our validation process. We also warmly thank the reviewers for their very valuable feedback. This work received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101021607 and the last author acknowledges the support of the French Research Agency via the ANR ParSiTi project (ANR16-CE33-0021).

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Omer Arshad, Ignazio Gallo, Shah Nawaz, and Alessandro Calefati. 2019. Aiding intra-text representations with visual context for multimodal named entity recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 337–342. IEEE.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.

Ana-Maria Bucur, Adrian Cosma, and Ioan-Bogdan Iordache. 2022. Blue at memotion 2.0 2022: You have my image, my text and my transformer. *arXiv preprint arXiv:2202.07543*.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, pages 152–164.

Shuguang Chen, Gustavo Aguilar, Leonardo Neves, and Thamar Solorio. 2021. Can images help recognize entities? a study of the role of images for multimodal NER. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 87–96, Online. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Patrick Davison. 2012. *9. The Language of Internet Memes*, pages 120–134. New York University Press.

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.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624. Curran Associates, Inc.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Lambert Mathias, Shaoliang Nie, Aida Mostafazadeh Davani, Douwe Kiela, Vinodkumar Prabhakaran, Bertie Vidgen, and Zeerak Waseem. 2021. Findings of the WOAH 5 shared task on fine grained hateful memes detection. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 201–206, Online. Association for Computational Linguistics.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Martina Miliani, Giulia Giorgi, Ilir Rama, Guido Anselmi, and Gianluca E Lebani. 2020. Dankmemes@ evalita 2020: The memeing of life: Memes, multimodality and politics. In *EVALITA*.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarah Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. 2020. Grounded situation recognition. In *European Conference on Computer Vision*, pages 314–332. Springer.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

David Samuel and Milan Straka. 2021. ÚFAL at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 483–492, Online. Association for Computational Linguistics.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 task 8: Memotion analysis- the visuolingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Shivam Sharma, Tharun Suresh, Atharva Kulkarni, Himanshi Mathur, Preslav Nakov, Md. Shad Akhtar, and Tanmoy Chakraborty. 2022. Findings of the constraint 2022 shared task on detecting the hero, the villain, and the victim in memes. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations - CONSTRAINT 2022, Collocated with ACL 2022*.

Carina Silberer and Manfred Pinkal. 2018. Grounding semantic roles in images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2616–2626, Brussels, Belgium. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework.

Bernard Lewis Welch. 1947. The generalization of 'student's' problem when several different population variances are involved. *Biometrika*, 34(1-2):28–35.

Hao Yang, Hao Wu, and Hao Chen. 2019. Detecting 11k classes: Large scale object detection without fine-grained bounding boxes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9805–9813.

A Human Annotations

To assess the quality of the dataset and put our results into perspective, we hand labeled part of the datasets. The team of five annotators is composed of researchers in Natural Language Processing. One of them is American native and the other 4 are European. Two of them are in the 40-50s age range and three of them are in the 20-30s. The annotators were all given the same 100 samples to label. To have a better estimate of the macro-F_1, we sampled 25 memes for each gold role. The annotator were given the class definitions and were informed that the labels had a uniform distribution. The annotation script as well as the answers of the annotators are available with the remainder of our code at `https://github.com/smontariol/mmsrl_constraint`.

We compute the macro-F_1 score of each annotator, resulting in an average score of 0.65. The minimum score was 0.57 and the maximum 0.69.

These scores show the difficulty of the task for a human. For comparison, the best score during the challenge was 0.58, still considerably lower than the human best score.

To measure the inter-annotator agreement, we compute the average pair-wise Cohen's κ (Cohen, 1960). It is similar to measuring the percentage of agreement, but taking into account the possibility of the agreement between two annotators to occur by chance for each annotated sample. The average Cohen's κ is 0.47, indicating a "moderate" agreement according to Cohen (1960). However, it also indicates that less than one third of the annotations are reliable (McHugh, 2012).