# SESAM at SemEval-2020 Task 8: Investigating the relationship between image and text in sentiment analysis of memes

**Lisa Bonheme**
School of Computing
University of Kent
Canterbury, CT2 7NF, UK
lb732@kent.ac.uk

**Marek Grzes**
School of Computing
University of Kent
Canterbury, CT2 7NF, UK
m.grzes@kent.ac.uk

## Abstract

This paper presents our submission to task 8 (memotion analysis) of the SemEval 2020 competition. We explain the algorithms that were used to learn our models along with the process of tuning the algorithms and selecting the best model. Since meme analysis is a challenging task with two distinct modalities, we studied the impact of different multimodal representation strategies. The results of several approaches to dealing with multimodal data are therefore discussed in the paper. We found that alignment-based strategies did not perform well on memes. Our quantitative results also showed that images and text were uncorrelated. Fusion-based strategies did not show significant improvements and using one modality only (text or image) tends to lead to better results when applied with the predictive models that we used in our research.

## 1 Introduction

**SemEval 2020 task 8 (Sharma et al., 2020)** is a sentiment analysis task targeted at memes[1] divided into three sub-tasks of increasing complexity: **Sub-task A** is predicting the sentiment polarity of a meme, **Sub-task B** is a multi-label binary classification task which aims to predict whether a meme is humorous, offensive, sarcastic and/or motivational (it can also have neither of these attributes), **Sub-task C** is a multi-output ordinal classification task which aims to predict the degree of humour, offence, sarcasm and motivation of a meme.

The dataset used for this task contains memes images whose text has been extracted by optical character recognition (OCR) and manually corrected when needed. Each meme is annotated on different aspects: sentiment polarity for sub-task A and the degree of humour, sarcasm, offence and motivation for sub-tasks B and C.

Memes sentiment analysis is a challenging task as memes are multi-modal, rely heavily on implicit knowledge, and often use humour and sarcasm. While this topic is of growing interest for NLP community, the way image and text interact in memes has barely been explored, leading to sub-optimal representation learning.

In an attempt to shed some light on the role of both modalities, we investigate their correlation and their impact on each sub-task prediction. Our code is available at https://github.com/bonheml/SESAM.

## 2 Related work

Sentiment analysis of text is a very active research area which still faces multiple challenges such as irony and humour detection (Hernández Farias and Rosso, 2017) and low inter-annotator agreement caused by the high subjectivity of the content (Mohammad, 2017).

Research has been extended to multimodal sentiment analysis during the last years (Soleymani et al., 2017), but the focus was mostly on video and text or speech and text. The specific multi-modality of memes in sentiment analysis has only been addressed recently by French (2017), who investigated their correlation with other comments in online discussions.

[1] We use the term meme to refer to internet memes as defined in Davidson (2012). The memes considered in this task are only composed of image and text.

The growing usage of memes as an alternative medium of communication on social media has also recently drawn the attention of the online abuse research community. Zannettou et al. (2018) studied the propagation of memes posted by fringe web communities[2], and their influence and transmission between different social media. Sabat et al. (2019) performed hate speech detection on memes and showed that images were more important than text for the prediction.

However, as pointed out by Vidgen et al. (2019), memes completely make sense only if one takes both text and image content into account. These modalities can also lead to totally different perceived sentiment when recombined. For example, a meme whose image is a grumpy cat and the text is "happy birthday" will have a very different sentiment from a meme with the same text but with an image of a happy puppy.

We argue that having a better understanding of both modalities interaction will contribute to more informed joint representations and is a crucial topic to explore.

Thus, we investigate the impact of multiple embeddings applied to both modalities on several models with different types of decision boundaries and verify the consistency of our findings assessing the embeddings across the three different sub-tasks. As our main focus is to study the impact of representations used, we chose simple classification models from Scikit-learn (Pedregosa et al., 2011) such as K nearest neighbours or Gaussian Naïve Bayes over more complex ones such as the deep learning architecture composed of three bidirectional gated recurrent units networks with contextual intermodal attention proposed by Akhtar et al. (2019). We did not perform any hyper-parameter tuning.

In multi-view representation learning, different techniques can be used to represent the views, depending on the nature of the relationship between them (Guo et al., 2019). When they share latent traits, one can use alignment to project their embeddings into a common space given a constraint (e.g., distance, correlation) (Baltrušaitis et al., 2019). On the other hand, if they are complementary, fusion techniques will be more useful as they will group the meaningful latent variables of each view into a compact representation (Li et al., 2019).

In section 3.1 we assess the usefulness of aligned representation for memes by investigating the possible correlations between images and memes. Then, in section 3.2, we study the added value of voter-based fusion techniques such as the one proposed by Gaspar and Alexandre (2019) in their work on multimodal sentiment analysis.

### 2.1 How correlated are images and text?

Exploring the possible correlations between images and text can provide valuable insights into the efficiency of the aligned representation for memes. Canonical correlation analysis (CCA) (Hotelling, 1936) has proven to be very efficient for correlation-based multimodal representation learning alignment (Wang et al., 2015), and has been successfully used for cross-modal multimedia retrieval (Rasiwasia et al., 2010). In order to provide a broad analysis of correlation, we analyse both linear and non-linear relationships between image and text embeddings.

**Linear CCA** Introduced by Hotelling (1936), CCA aims to find the linear projections of two views which are maximally correlated.

More formally, let $X \in R^{n \times m}$ and $Y \in R^{n \times p}$ be two zero-mean matrices of $n$ observations with $m$ and $p$ features respectively. We aim to find the K orthogonal linear projections $A = [a_1, \ldots, a_k]$, $B = [b_1, \ldots, b_k]$ such that:

$$(a^*, b^*) = \underset{a,b}{\operatorname{argmax}} \operatorname{corr}(a^T X, b^T Y) = \underset{a,b}{\operatorname{argmax}} \frac{a^T \Sigma_{XY} b}{\sqrt{a^T \Sigma_{XX} a} \sqrt{b^T \Sigma_{YY} b}}$$

where $\Sigma_{XX}$ and $\Sigma_{YY}$ are the covariance matrices of X and Y respectively and $\Sigma_{XY}$ is their cross covariance matrix (Uurtio et al., 2017).

[2] We refer to fringe web communities as online communities sharing a deviant subculture, similarly to what is described by Zannettou et al. (2018)

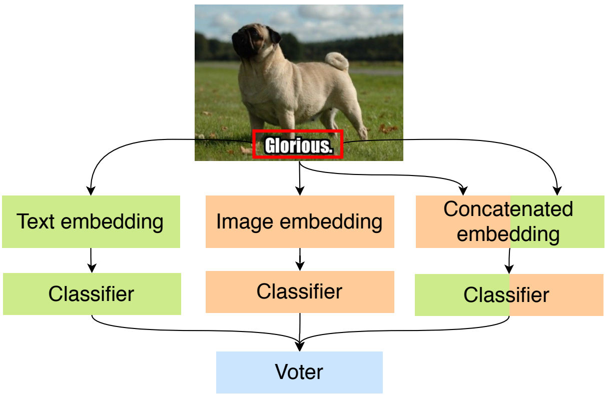

Figure 1: Overview of the voter method

**Deep CCA** As standard CCA is only able to discover linear relationships, other techniques such as Kernel CCA (KCCA) (Lai and Fyfe, 2000; Melzer et al., 2001; Van Gestel et al., 2001; Akaho, 2001) have been developed to discover non-linear associations. However, KCCA does not scale well to large datasets. Using the better scaling capacity of deep neural networks, Andrew et al. (2013) proposed deep CCA (DCCA), a version of CCA which stacks layers of non-linear transformations for both views and optimises the correlation between their transformed representations. Given the size of the dataset and the high dimensionality of the features used in this study, we chose to use DCCA over KCCA.

**Application to the tasks** Both CCA and DCCA are applied to the training dataset. CCA results are evaluated using the first canonical correlation scores and the assessment of their statistical significance. As DCCA provides only aligned embeddings, it cannot be evaluated using the same techniques. Instead, we trained DCCA on the training dataset, predicted the aligned embeddings of the dev and testing dataset and compared the results of our different models, discussed in section 2.2, with aligned and non-aligned embeddings. We also investigated the intra-class correlation by performing CCA on each class of sub-task A and each label of sub-task B.

### 2.2 How image and text contribute to the predictions?

**Fusion methods** Over the years, various fusion techniques for predictive models have been developed. Some rely on a neural network to perform the fusion (Tanti et al., 2017), or just concatenate the modalities into one vector and treat it as a unimodal problem (Baltrušaitis et al., 2019). However, it is difficult to uncover the contribution of each modality with these techniques. In contrast, a voter-based fusion technique (Morvant et al., 2014; Gaspar and Alexandre, 2019) can be easily interpreted and will thus be used here. This technique is referred to as *late fusion* as the fusion is performed after the learning phase whereas techniques such as embedding concatenation, where the fusion occurs before the learning phase, are referred to as *early fusion*.

As voter fusion is model-agnostic, it also allows us to test it on different models and tasks to verify the generalisation of our findings. While late fusion has been shown to often provide better results in multimedia fusion (Snoek et al., 2005), early fusion tends to perform better when one of the modalities contribute more than the other to the predictions (Morvant et al., 2014).

To handle this possibility, our voter, illustrated in figure 1, is composed of three identical models which are trained on image, text and a concatenation of both embeddings respectively. Thus, we perform *hybrid fusion*, using the information provided by both late and early fusion. As we are only interested in exploring the impact of the different modalities, unlike in Gaspar and Alexandre (2019) where classifier decisions were weighted according to their quality, we gave all classifiers the same weights. To assess whether different modalities contribute to a different type of prediction, we also run each model independently and compare their results. Thus, if a modality is only helpful in some cases (e.g., only for negative polarity detection), the voter should provide better results than independent models.

| Sub-task | Type of classification | Meta classifiers |
|---|---|---|
| A | Ordinal | Ordinal (Frank and Hall, 2001) |
| B | Multi-label binary | One versus Rest |
| C | Multi-output ordinal | Multi-output and Ordinal |

Table 1: Meta classifiers used for the different sub-tasks

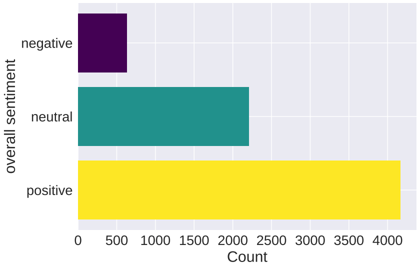

Figure 2: Polarity distribution of sub-task A (training dataset)

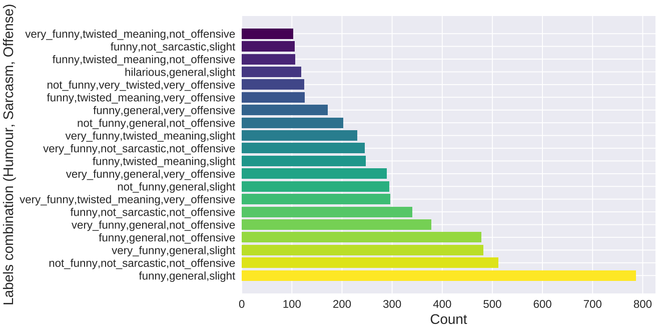

Figure 3: Labels distribution of sub-task C (training dataset)

**Models** The different predictive models used are logistic regression (LR), K nearest neighbours (KNN), Gaussian Naïve Bayes (GNB), Random forest (RF), and multi-layer perceptron (MLP). We chose them so that we can study the impact of different embeddings on several decision boundaries. To suit the different sub-tasks objectives, these models are wrapped in meta classifiers as described in table 1. We used implementation from Scikit-learn (Pedregosa et al., 2011) for the multi-output and multi-label classifiers and a custom implementation of Frank and Hall (2001) made compatible with Scikit-learn models for the ordinal classifier.

## 3 Experimental setup

**Data cleaning and preprocessing** We manually added the text values of seven memes which had neither OCR nor corrected text values in the training dataset and removed URLs corresponding to meme sources from transcribed texts. A number of websites used for meme generation add their URL to the final meme, and this was often caught and transcribed by the OCR extraction. Following Camacho-Collados and Pilehvar (2018) study on text preprocessing, we did not perform any lemmatisation or lowercasing. To obtain one text embedding per meme, the text of each meme was vectorised using a pre-trained Universal Sentence Encoder (USE) (Cer et al., 2018) retrieved from Tensorflow hub (Abadi et al., 2015). The images were processed with Xception (Chollet, 2017), pre-trained on ImageNet (Russakovsky et al., 2015), and the penultimate layer was used as embedding.

**Dataset analysis** As shown in figures 2 and 3, the training dataset is highly skewed towards positive memes which are mostly funny, motivational, slightly sarcastic and offensive. Occurrences of “extreme” memes such as hateful offensive are very rare (e.g., there are less than 500 hateful offensive memes). The word count distribution is equivalent over each label, and we did not find words specifically attached to a given label. No obvious cluster of memes was shown by the t-SNE (van der Maaten and Hinton, 2008) or UMAP (McInnes et al., 2018) projections of sentence and image embeddings.

**Models training** The same model types and embeddings combinations are used for the three sub-tasks, and we only varied the meta classifiers as listed in table 1 to adapt the models’ output to the task at hand.

**Experiment** Model training and evaluation was performed in three phases:

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| LR | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 |
| KNN | 0.45 | 0.32* | 0.47 | 0.43 | 0.46 |
| GNB | 0.39 | 0.20 | 0.20 | 0.21 | 0.33 |
| RF | 0.88 | 0.33 | 0.32 | 0.52 | 0.32 |
| MLP | 0.86 | 0.33 | 0.73 | 0.77 | 0.25 |

Table 2: Macro F1 scores for sub-task A on the dev dataset. The score from the model selected during the evaluation phase to be submitted to the competition is marked with *

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| LR | 0.62 | 0.62 | 0.62 | 0.62 | 0.62 |
| KNN | 0.72 | 0.65* | 0.73 | 0.71 | 0.73 |
| GNB | 0.67 | 0.38 | 0.38 | 0.38 | 0.58 |
| RF | 0.93 | 0.64 | 0.66 | 0.78 | 0.66 |
| MLP | 0.92 | 0.64 | 0.84 | 0.88 | 0.62 |

Table 3: Averaged macro F1 scores for sub-task B on the dev dataset. The score from the model selected during the evaluation phase to be submitted to the competition is marked with *

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| LR | 0.28 | 0.21 | 0.29 | 0.22 | 0.28 |
| KNN | 0.50 | 0.29* | 0.57 | 0.41 | 0.55 |
| GNB | 0.45 | 0.25 | 0.35 | 0.28 | 0.40 |
| RF | 0.94 | 0.31 | 0.54 | 0.57 | 0.59 |
| MLP | 0.93 | 0.30 | 0.82 | 0.78 | 0.45 |

Table 4: Averaged macro F1 scores for sub-task C on the dev dataset. The score from the model selected during the evaluation phase to be submitted to the competition is marked with *

**The training phase** where a training and a dev dataset are provided. During this phase, each architecture[3] is trained on the training dataset and evaluated on the dev dataset using the macro F1 score for sub-task A, and averaged macro F1 scores for sub-tasks B and C[4]. No hyper-parameter tuning is performed and the dev dataset is only used to filter non-informative models which will not be submitted during the evaluation phase.

**The evaluation phase** where an unlabelled testing dataset is provided. During this phase, the predictions are done using the architectures previously selected, without retraining, and uploaded to Codalab. Similarly to the training phase, the results are evaluated using a macro F1 score for sub-task A and an averaged macro F1 scores for sub-tasks B and C. The combination of model type and embedding providing the best results on the testing dataset over the three sub-tasks is selected for the final ranking.

**The ranking phase** where the model selected during the evaluation phase is submitted for ranking. The final ranking is done using the testing dataset and the same metrics as in the previous phases.

## 4 Results

This section provides an analysis of our results at each step of our experiment. First, we investigate the results on the dev dataset which guided our model selection for the evaluation phase. The results retrieved from Codalab during the evaluation phase for the selected models are then analysed, and we finally conclude with the analysis of the scores provided during the final ranking.

### 4.1 Evaluation on dev dataset

**Alignment approach (correlation-based)** No statistically significant correlation between image and text was found with linear CCA, either over the entire training dataset or intra-class. Non-linear Deep CCA (DCCA) does not provide significant improvements compared to non-aligned concatenated embeddings or text modality only, and even often worsened the results. It thus seems that image and text are more complementary than correlated in the case of sentiment analysis of memes. This finding is consistent with the fact that memes often make sense when the combination of image and text is considered and changing one or the other can change the associated sentiments (Vidgen et al., 2019). Given these empirical results, we conclude that alignment-based approaches may not be suitable for meme analysis.

**Fusion approach (voter-based)** As shown in tables 2, 3 and 4, voter-based fusion technique did not lead to better results than one modality alone and consistently worsened the results. Interestingly, all the models tested, except KNN, performed better with one modality alone over all the sub-tasks.

Most of the models were not able to find very discriminative features in image embeddings and often ended up predicting every class as belonging to the most frequent class. This problem was also reflected in concatenated embeddings whose results were most of the time worse than the ones provided by the most informative modality. While early and hybrid fusion approaches (i.e., concatenated embeddings and voters) provide better scores than image-only, text-only generally gives the best results, especially for GNB, RF and MLP.

While it intuitively makes sense to consider images in memes, it seems that image representations such as the one we used may not be suitable for the task at hand and thus underperformed. Indeed, these representations are accurate enough to perform image captioning, but they lack the higher-level information we use to interpret memes. For example, a surprised cat and a grumpy cat will just be represented as cats when the sentiment attached to a meme "Me when I look at my grades" can drastically change depending on the type of cat used. Thus, using embeddings extracted from image sentiment classifiers could be more suitable to sentiment analysis of memes.

Because of the reuse of the same image with a different text leading to different sentiments, using image embeddings only can also introduce noise to the data with one image linked to contradictory outputs. Thus, it may be more efficient to merge both embeddings early on. However, early fusion did not show consistent improvements, indicating that more complex, model-dependent fusion techniques such as neural networks may be needed.

Except for KNN that obtained marginally better results with fusion techniques, most models seem to perform best with text, contrary to what was reported by Sabat et al. (2019) for hate speech detection. These apparently contradictory results may be due to the usage of different discriminative features on each task. This could be an interesting avenue to explore for assessing the potential of transfer learning with memes embeddings. Indeed, the more different discriminative features used for sentiment analysis and hate speech detection of memes are, the less efficient the usage of generic meme embeddings will be.

Finally, as memes often reflect the shared culture of the communities which create them (Lin et al., 2014), having some contextual knowledge would probably also be greatly beneficial for meme analysis.

**Model selection** Given that the results of linear regression are very low and do not provide much information on the impact of each modality, it is removed from the pool of models that will be used for the evaluation phase.

### 4.2 Evaluation on the testing dataset

During the evaluation phase, we used the results provided on Codalab, which are referenced in tables 5, 7 and 9 to select the model to submit for competition ranking and assess the generalisability of the conclusion made from the empirical results during the training phase. After the release of the final ranking,

[3] What we referred to as architecture here is the combination of a given model type, a given embedding and a given meta classifier (e.g., KNN with text embeddings and ordinal meta-classifier)

[4] The higher the scores, the better the model. Macro F1 and averaged macro F1 metrics are detailed in the task description (Sharma et al., 2020)

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| KNN | 0.34 | 0.35* | 0.32 | 0.32 | 0.34 |
| GNB | 0.34 | 0.20 | 0.20 | 0.21 | 0.30 |
| RF | 0.26 | 0.27 | 0.25 | 0.25 | 0.26 |
| MLP | 0.33 | 0.31 | 0.32 | 0.33 | 0.25 |
| Baseline |  |  | 0.22 |  |  |

Table 5: Macro F1 scores originally provided for sub-task A on the testing dataset of the evaluation phase. The score from the model submitted to the competition is marked with *

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| KNN | 0.34 | 0.35* | 0.32 | 0.32 | 0.34 |
| GNB | 0.34 | 0.20 | 0.20 | 0.21 | 0.30 |
| RF | 0.26 | 0.27 | 0.25 | 0.25 | 0.26 |
| MLP | 0.33 | 0.33 | 0.33 | 0.33 | 0.25 |
| Baseline |  |  | 0.22 |  |  |

Table 6: Corrected macro F1 scores provided after the competition for sub-task A on the testing dataset of the evaluation phase. The score from the model submitted to the competition is marked with *

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| KNN | 0.66 | 0.67* | 0.66 | 0.66 | 0.66 |
| GNB | 0.63 | 0.36 | 0.36 | 0.36 | 0.60 |
| RF | 0.64 | 0.63 | 0.63 | 0.63 | 0.64 |
| MLP | 0.65 | 0.64 | 0.65 | 0.66 | 0.63 |
| Baseline |  |  | 0.51 |  |  |

Table 7: Averaged macro F1 scores originally provided for sub-task B on the testing dataset of the evaluation phase. The score from the model submitted to the competition is marked with *

|  | Score per type of embedding |  |  |  |  |
|---|---|---|---|---|---|
|  | Text | Image | Concat | Voter | DCCA |
| KNN | 0.49 | 0.49* | 0.49 | 0.46 | 0.49 |
| GNB | 0.52 | 0.37 | 0.37 | 0.37 | 0.48 |
| RF | 0.43 | 0.43 | 0.42 | 0.41 | 0.44 |
| MLP | 0.51 | 0.49 | 0.50 | 0.50 | 0.41 |
| Baseline |  |  | 0.51 |  |  |

Table 8: Corrected averaged macro F1 scores provided after the competition for sub-task B on the testing dataset of the evaluation phase. The score from the model submitted to the competition is marked with *

the task organisers have notified us that the scores displayed on Codalab during the competition were incorrect and have released corrected scores which we discuss in section 4.3. In this section, we discuss the incorrect Codalab evaluation scores because this is what was available to us during the competition, and we used these to select the final model that we submitted.

**Alignment approach (correlation-based)** Similarly to the results observed during the training phase, no statistically significant correlation between image and text was found with linear CCA. The scores obtained with DCCA were also lower than the ones obtained during the training phase. Thus, we did not use the architecture with aligned embeddings.

**Fusion approach (voter-based)** Surprisingly, the results in the evaluation phase were very different from those obtained during the training phase. RF and MLP, which were both performing very well on text modality over all three sub-tasks had consistently lower scores with almost equivalent results over all the embeddings tested. KNN which was previously performing well on fusion-based embeddings also provided lower scores which were almost equivalent over all the embeddings tested, with a marginal improvement with image embeddings.

**Model selection** The model best performing on the testing dataset of the evaluation phase, KNN with image embedding, was the one submitted for the final ranking. When this analysis was performed, we did not know that the results in tables 5, 7, 9 were incorrect.

### 4.3 Final results

In this section, we present the correct evaluation results which became available after the competition.

| | Score per type of embedding | | | | |
|---|---|---|---|---|---|
| | Text | Image | Concat | Voter | DCCA |
| KNN | 0.26 | 0.26* | 0.26 | 0.23 | 0.25 |
| GNB | 0.29 | 0.26 | 0.26 | 0.27 | 0.23 |
| RF | 0.18 | 0.18 | 0.17 | 0.16 | 0.19 |
| MLP | 0.27 | 0.26 | 0.28 | 0.25 | 0.12 |
| Baseline | | | 0.25 | | |

Table 9: Averaged macro F1 scores originally provided for sub-task C on the testing dataset of the evaluation phase. The score from the model submitted to the competition is marked with *

| | Score per type of embedding | | | | |
|---|---|---|---|---|---|
| | Text | Image | Concat | Voter | DCCA |
| KNN | 0.31 | 0.31* | 0.30 | 0.30 | 0.30 |
| GNB | 0.32 | 0.23 | 0.23 | 0.25 | 0.28 |
| RF | 0.27 | 0.27 | 0.17 | 0.27 | 0.27 |
| MLP | 0.31 | 0.31 | 0.31 | 0.31 | 0.22 |
| Baseline | | | 0.25 | | |

Table 10: Corrected averaged macro F1 scores provided after the competition for sub-task C on the testing dataset of the evaluation phase. The score from the model submitted to the competition is marked with *

**Alignment approach (correlation-based)** Similarly to what was observed during the training phase, DCCA did not provide any significant improvements compared to other embeddings and have worsened the results most of the time. Given the consistency of the results, we can conclude that correlation-based alignment techniques are not useful for sentiment analysis of memes as, unlike image captions, memes show no evidence of correlation between image and text.

**Fusion approach (voter-based)** As shown in tables 6, 8 and 10, the corrected results are very close to the original evaluation results for sub-task A, but vary importantly for sub-tasks B and C. Similarly to the results obtained during the training phase, GNB still favours text-only modality for all the tasks, but other models now show similar results for text, image, concat and voter. Interestingly, text embeddings provide much lower results than during the training phase, especially for RF and MLP. Various factors such as different label distributions between dev and testing dataset, more similar vocabulary between dev and training dataset, or memes with less informative text in the testing dataset could have influenced these results. We argue that an in-depth analysis of these possible factors could lead to new insights regarding the embedding features used by the models during the learning process. Thus, we will investigate it once the annotated testing dataset has been published.

**Model selection** Given the corrected scores, GNB with text embeddings would have been a better model to submit for final ranking, especially for sub-task B. Unfortunately, the correct evaluation scores were not available during the competition.

## 5 Conclusion

We have provided an overview of the impact of different representations on meme sentiment analysis. We tested alignment-based and fusion-based techniques on a range of models on each sub-task. While none of them seemed to be beneficial for the different sub-tasks, we found that 1) alignment-based techniques were not suitable for meme analysis as image and text of memes are not correlated 2) using only one modality (text or image) tends to perform better than a combination of both when we use standard (i.e. non-deep learning) predictive models. However, these conclusions should be taken with caution as the scores obtained on the dev and testing datasets vary greatly and other factors, such as the label distribution of the dataset can also have influenced these results. Finally, we argue that a more adapted image representation, possibly enriched with contextual knowledge, as well as more complex fusion techniques, may be promising to explore.

**Acknowledgement** We thank the SemEval-2020 organisers for their time to prepare the data and run the competition, and the reviewers for their insightful comments.

## References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Shotaro Akaho. 2001. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Springer-Verlag.

Md Shad Akhtar, Dushyant Chauhan, Deepanway Ghosal, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multi-task Learning for Multi-modal Emotion Recognition and Sentiment Analysis. 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 370–379, Minneapolis, Minnesota. Association for Computational Linguistics.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep Canonical Correlation Analysis. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA. PMLR.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, February.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 40–46, Brussels, Belgium. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November. Association for Computational Linguistics.

Francois Chollet. 2017. Xception: Deep Learning With Depthwise Separable Convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July.

Patrick Davidson. 2012. The language of internet memes. In *The social media reader*, chapter 9, pages 120–134. New York University Press, New York, New York, USA.

Eibe Frank and Mark Hall. 2001. A Simple Approach to Ordinal Classification. In Luc De Raedt and Peter Flach, editors, *Machine Learning: ECML 2001*, pages 145–156, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jean H. French. 2017. Image-based memes as sentiment predictors. In *2017 International Conference on Information Society (i-Society)*, pages 80–85, July.

António Gaspar and Luís A. Alexandre. 2019. A Multimodal Approach to Image Sentiment Analysis. In Hujun Yin, David Camacho, Peter Tino, Antonio J. Tallón-Ballesteros, Ronaldo Menezes, and Richard Allmendinger, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, pages 302–309, Cham. Springer International Publishing.

Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. Deep Multimodal Representation Learning: A Survey. *IEEE Access*, 7:63373–63394.

Delia I. Hernández Farias and Paulo Rosso. 2017. Chapter 7 - Irony, Sarcasm, and Sentiment Analysis. In Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Bing Liu, editors, *Sentiment Analysis in Social Networks*, pages 113–128. Morgan Kaufmann, Boston.

Harold Hotelling. 1936. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321, December.

Pei Ling Lai and Colin Fyfe. 2000. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377.

Yingming Li, Ming Yang, and Zhongfei (Mark) Zhang. 2019. A Survey of Multi-View Representation Learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1863–1883, October.

Chi-Chin Lin, Yi-Ching Huang, and Jane Yung-jen Hsu. 2014. Crowdsourced explanations for humorous internet memes based on linguistic theories. In Jeffrey P. Bigham and David C. Parkes, editors, *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA*. AAAI.

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. February.

Thomas Melzer, Michael Reiter, and Horst Bischof. 2001. Nonlinear Feature Extraction Using Generalized Canonical Correlation Analysis. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, *Artificial Neural Networks — ICANN 2001*, pages 353–360, Berlin, Heidelberg. Springer Berlin Heidelberg.

Saif M. Mohammad. 2017. Challenges in Sentiment Analysis. In Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco, editors, *A Practical Guide to Sentiment Analysis*, number Section 7, pages 61–83. Springer International Publishing, Cham.

Emilie Morvant, Amaury Habrard, and Stéphane Ayache. 2014. Majority Vote of Diverse Classifiers for Late Fusion. In Pasi Fränti, Gavin Brown, Marco Loog, Francisco Escolano, and Marcello Pelillo, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 153–162, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 25–260, New York, NY, USA. Association for Computing Machinery.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Benet Oriol Sabat, Cristian Canton Ferrer, and Xavier Giro-i-Nieto. 2019. Hate Speech in Pixels: Detection of Offensive Memes towards Automatic Moderation. In *AI for Social Good Workshop NEURIPS2019*.

Chhavi Sharma, William Paka, Scott, Deepesh Bhageria, Amitava Das, Soujanya Poria, Tanmoy Chakraborty, and Björn Gambäck. 2020. Task report: Memotion analysis 1.0 @semeval 2020: The visuo-lingual metaphor! In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, September. Association for Computational Linguistics.

Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA'05, pages 399–402, New York, NY, USA. Association for Computing Machinery.

Mohammad Soleymani, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. 2017. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14.

Marc Tanti, Albert Gatt, and Kenneth Camilleri. 2017. What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator? In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 51–60, Santiago de Compostela, Spain, September. Association for Computational Linguistics.

Viivi Uurtio, João M. Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. 2017. A Tutorial on Canonical Correlation Methods. *ACM Computing Surveys*, 50(6):1–33, November.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Tony Van Gestel, Johan A K Suykens, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. 2001. Kernel Canonical Correlation Analysis and Least Squares Support Vector Machines. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, *Artificial Neural Networks — ICANN 2001*, pages 384–389, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93. Association for Computational Linguistics.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1083–1092, Lille, France, July. PMLR.

Savvas Zannettou, Tristan Caulfield, Jeremy Blackburn, Emiliano De Cristofaro, Michael Sirivianos, Gianluca Stringhini, and Guillermo Suarez-Tangil. 2018. On the Origins of Memes by Means of Fringe Web Communities. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 188–202, New York, NY, USA. ACM.

# A Hyper-parameters of the models

In addition to the code provided at `https://github.com/bonheml/SESAM`, the language, packages used and their version, and the hyper-parameters of the models are detailed in this appendix for reproducibility.

## A.1 Language and library used

The experiment was implemented in Python 3.6 and the packages listed in table 11 were installed with the package manager Pip.

| Package | Version | Usage |
|---|---|---|
| Pandas | 0.25.3 | Data analysis |
| Matplotlib | 3.1.2 | Data visualisation |
| Seaborn | 0.9.0 | Data visualisation |
| Wordcloud | 1.6.0 | Data visualisation |
| Jupyter | 1.0.0 | Notebook |
| Numpy | 1.17.4 | CCA implementation |
| Scipy | 1.4.1 | CCA implementation |
| Tensorflow | 2.1.0 | DCCA implementation and pre-trained Xception download |
| Tensorflow-hub | 0.7.0 | Pre-trained USE download |
| Scikit-learn | 0.21.3 | Models and metrics |

Table 11: Packages used for the experiment.

## A.2 Hyper-parameters of the models

This section details the hyper-parameters of each model. The complete model training can also be viewed as a notebook at `https://github.com/bonheml/SESAM/blob/master/models_training.ipynb` for Scikit-learn models, and `https://github.com/bonheml/SESAM/blob/master/deep_cca.ipynb` for DCCA.

**Logistic regression:** Stratified K-fold (5 folds), random seed of 0, Saga solver, maximum of 10000 iterations, 6 CPU jobs, other parameters are default values from Scikit-learn (see `https://scikit-learn.org/0.21/modules/generated/sklearn.linear_model.LogisticRegressionCV.html`).

**K nearest neighbours:** 6 CPU jobs, other parameters are default values from Scikit-learn (see `https://scikit-learn.org/0.21/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`).

**Gaussian Naïve Bayes:** Parameters are default values from Scikit-learn (see `https://scikit-learn.org/0.21/modules/generated/sklearn.naive_bayes.GaussianNB.html`)

**Random forest:** Random seed of 0, generalization accuracy estimated with out-of-bag samples, 6 CPU jobs, other parameters are default values from Scikit-learn (see `https://scikit-learn.org/0.21/modules/generated/sklearn.ensemble.RandomForestClassifier.html`).

**Multi-layer perceptron:** Maximum of 1000 iterations, other parameters are default values from Scikit-learn (see `https://scikit-learn.org/0.21/modules/generated/sklearn.neural_network.MLPClassifier.html`).

**DCCA:** The model is composed of 3 densely-connected layers of 1000 units with sigmoid activation and an output layer of 100 units with linear activation. It is trained using a batch size of 800 during 100 epochs using all singular values. The model is optimised with RMSprop using a learning rate of 1e-3 and an L2 penalty of 1e-5. The complete implementation is available at `https://github.com/bonheml/SESAM/blob/master/src/models/dcca.py`.