

Do we need Multimodality? Experiments with Tweets from European Union Executives

Sina Özdemir¹, Patrick Schwabl²,

¹Department of Sociology and Political Science, Norwegian University of Science and Technology

²Department of Media and Communication, LMU Munich

Author Note

Patrick Schwabl  <https://orcid.org/0000-0001-7251-2919>

Correspondence concerning this article should be addressed to any of the two authors via Email.

Sina Özdemir: sina.ozdemir@ntnu.no

Patrick Schwabl: patrick.schwabl@ifkw.lmu.de

Introduction

Content analysis has always been one of the key methods in communication research and advances in computational methods often deal with processing vast quantities of text.

Yet, communication rarely happens via a single modality. For example, one of the key political actors in Europe, the European Union posts images in about 40% of its tweets (Özdemir & Rauh, 2022). Dictionary-based and shallow learning (SL) methods have a hard time incorporating multimodality into the analysis.

Deep learning (DL) brings the possibility to extend content analysis to multimodal materials. Previous studies have demonstrated the flexibility of embeddings to analyze multimodal data (Li et al., 2022; Niu et al., 2019; Tseng et al., 2021; Wu & Mebane, 2022).

In this paper, we evaluate the feasibility of using multimodal DL embeddings to classify political messages where the message is delivered with a combination of visual and textual modalities in a computational experiment. We build a series of unimodal SL models and multimodal DL embedding-based models to classify manually annotated tweets from European Union (EU) executives. We then compare the classification performance of these models. Our results indicate that multimodal signals are tricky to catch in a way that is meaningful to a classifier. Finally, we conclude with some recommendations for researchers who would like to use multimodal data in automated content analysis.

Research design and data

We use 898 manually annotated tweets from EU executive accounts posted between December 1st, 2019, and July 31st, 2020. Examples of them are presented in Figure 1. Tweets were manually coded as a whole based on whether they provide information on political operations, policies, programs, and reports published by the EU executive institutions. Overall 479 tweets contain such message (code: 1) and 419 tweets do not (code: 0). To ensure the data quality, three rounds of intercoder reliability tests were conducted between coders before coding the full sample (Krippendorff's $\alpha > .8$).

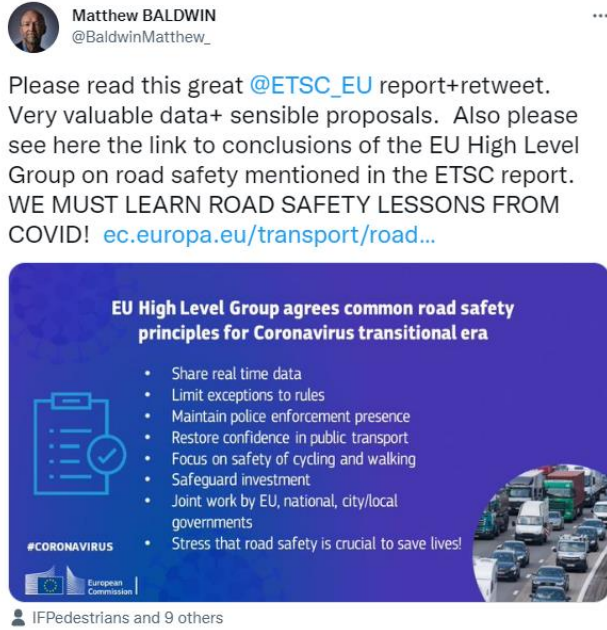


Figure 1: Example tweets

In our experiments, we take the binary indicator as the outcome and learn individual SL and DL predictive models. We use Quanteda (Benoit et al., 2018) for text preprocessing where we removed links and punctuation. For learning SL models we used the caRet (Kuhn, 2008) package in R and SciKitLearn (Pedregosa et al., 2011) in Python. For DL models, we utilize the transformers (Wolf et al., 2020) and Keras (Chollet, 2015) libraries in Python.

In our experimental setting, we test the predictive capacity of these models with three different ways of featurization. The first mode of featurization is a document feature matrix of the tweet texts with no weighting. The second is a term-frequency inverse-document-frequency (tf-idf) representation of tweet texts. Thirdly we combine textual and visual elements in a [898, 612, 768] embedding matrix as input data. This means, that for multimodal representations, there is a 612*768 feature matrix to represent every tweet and its image. When combining SL algorithms with embedding features we reduce the dimensionality of every tweet's feature matrix from two dimensions to one. We then reduce the number of features to two by performing principal component analysis on them.

This feature matrix is created using VisualBERT (Li et al., 2019). Figure 2 shows how VisualBERT combines textual and visual input into a combined feature matrix. The textual part is a BERT model, where word embeddings are generated from tokens. The visual embeddings are

generated using a pre-trained image model, that detects objects in images. In our case, we use Facebook's detectron2 library. (Wu et al., 2019) As seen in Figure 2 the image model produces region proposals and transforms these into embedding representations of those regions. Textual and visual embeddings are then passed into a transformer which is again of the same architecture as the BERT base version proposed by Devlin et al. (2019).

Training of VisualBERT happens by two tasks also closely related to the BERT training procedure. The authors call them (1) masked language modeling with the image and (2) sentence-image prediction. In (1) some textual elements from the text input are masked and the model must predict what that text should be. Image input is never masked. In (2) the model must decide which of two given captions belong to an image. This procedure is "allowing the model to implicitly discover useful alignments between both sets of inputs, and build up a new joint representation." (Li et al., 2019, p. 4)

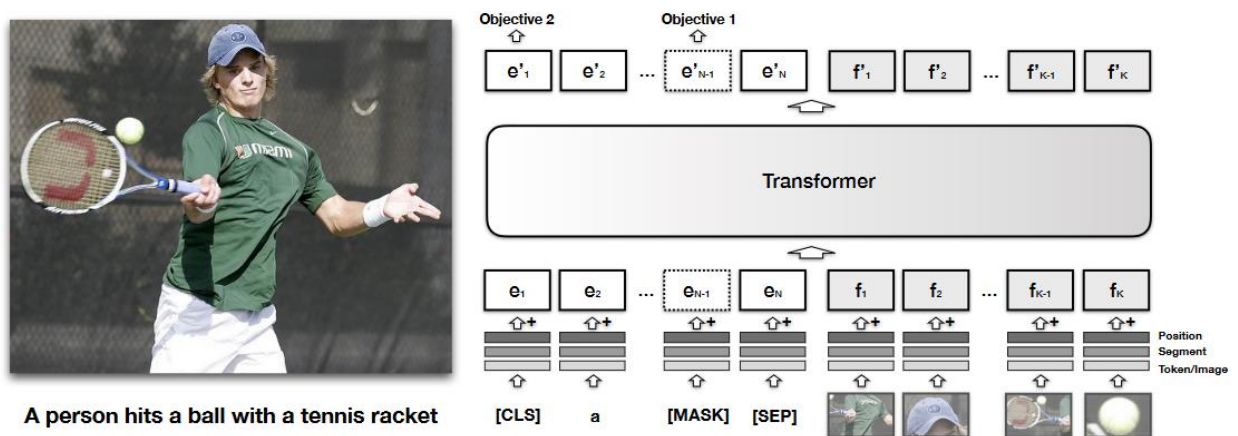


Figure 2: VisualBERT architecture, from L. H. Li et al. (2019)

Results

We report our experimental results on best-performing models with different featurization of text and image data in Table 1. Our experiments show three key results. First and foremost, text-only representations with SL models seem to outperform DL models with multimodal embeddings by a large margin. Our best-performing model is a random forest using tf-idf representation of text-only data. This is followed by random forest using a document-feature matrix. Our second key result is that SL models tend to perform better in terms of

precision, but not in recall, when the dataset is text only. The only exception to this is our SVM with tf-idf input where recall outweighs the precision by .10. We do not observe this pattern when the input dataset is multimodal embedding. Our last key result is that the multi-layer perceptron tends to outperform SL learners when data is represented by multimodal embeddings by about a margin of .10 in F1 score.

Table 1: Performance scores of predictive models

	DFM			TFIDF			Multimodal embeddings		
	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Naïve Bayes	0.57	0.48	0.70	0.37	0.51	0.29	-	-	-
Logistic regression	0.66	0.68	0.64	0.70	0.70	0.71	-	-	-
Support Vector machine	0.70	0.71	0.69	0.77	0.73	0.82	0.53	0.53	0.53
Random forest	0.72	0.75	0.69	0.78	0.78	0.77	0.52	0.52	0.52
XGBoost	0.67	0.64	0.70	0.71	0.72	0.70	-	-	-
MLP	-	-	-	-	-	-	0.62	0.63	0.62

Discussion and conclusion

Overall, our results indicate that it is not necessarily better to incorporate visual elements in text classification with automated analysis. This result begets insights for automated content analysis. While visual materials can be important for the delivery of the message, they may create more noise than signal in automated content analysis. In our case, the natural public relations visual content was extremely diverse often including creative visual infographics. These images, by nature, are rather different from training images often used in computer vision (CV) models.

Moreover, the substantive purposes of computer vision and social sciences often diverge from each other. Therefore, CV models tend to have a different purpose. This complicates choosing the right CV model even further. These circumstances call for visual material processing models tailored for social science purposes. While this would be a rather demanding task, it should be possible to create such models using existing machine learning architectures.

Based on our results, there are several good practices we can recommend for the future. First of all, DL algorithms require a large amount of data to reach acceptable performance. Therefore, it is always wise to start simple. As our results show, SL algorithms with simpler featurization can accomplish the task even if they do not encode information from visual materials. Therefore, our first recommendation is to test the simpler alternative. However, if the research question inextricably requires a multimodal analysis, we recommend two key actions. First of all, it is best to use DL algorithms with multimodal embeddings as they can handle high-dimension tensors better than SL algorithms. However, for this, we recommend researchers have a sufficient number of labeled observations for their model. Finally, for those researchers with limited resources, we would like to point out that it is still possible to use SL algorithms with multimodal embeddings. However, SL algorithms are not designed to handle tensors. This requires the researcher to apply dimension reduction to multimodal embeddings which may lead to substantial information loss as our experiments show. Therefore, it is imperative to find the best dimension reduction method before employing this option.

References

- Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30), 774. <https://doi.org/10.21105/joss.00774>
- Chollet, F. (2015). *Keras*. GitHub. <https://github.com/fchollet/keras>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. <http://arxiv.org/abs/1810.04805>

- Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5). <https://doi.org/10.18637/jss.v028.i05>
- Li, K., Zhang, Y., Li, K., Li, Y., & Fu, Y. (2022). Image-Text Embedding Learning via Visual and Textual Semantic Reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP. <https://doi.org/10.1109/TPAMI.2022.3148470>
- Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., & Chang, K.-W. (2019). VisualBERT: A Simple and Performant Baseline for Vision and Language. *ArXiv:1908.03557 [Cs]*. <http://arxiv.org/abs/1908.03557>
- Niu, Y., Lu, Z., Wen, J.-R., Xiang, T., & Chang, S.-F. (2019). Multimodal Multi-Scale Deep Learning for Large-Scale Image Annotation. *IEEE Transactions on Image Processing*, 28(4), 1720–1731. <https://doi.org/10.1109/TIP.2018.2881928>
- Özdemir, S., & Rauh, C. (2022). A Bird's Eye View: Supranational EU Actors on Twitter. *Politics and Governance*, 10(1), 133–145. <https://doi.org/10.17645/pag.v10i1.4686>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12, 2825–2830.
- Tseng, S.-Y., Narayanan, S., & Georgiou, P. (2021). Multimodal Embeddings From Language Models for Emotion Recognition in the Wild. *IEEE Signal Processing Letters*, 28, 608–612. <https://doi.org/10.1109/LSP.2021.3065598>
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., ... Rush, A. M. (2020). HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv:1910.03771 [Cs]*. <http://arxiv.org/abs/1910.03771>

Wu, P. Y., & Mebane, W. R. (2022). MARMOT: A Deep Learning Framework for Constructing Multimodal Representations for Vision-and-Language Tasks. *Computational Communication Research*, 4(1).

<https://doi.org/10.5117/CCR2022.1.008.WU>

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). *Detectron2*.

<https://github.com/facebookresearch/detectron2>