# Introduction:

Content analysis has always been one of the key methods in communication research. However, with the advent of web 3.0, the data sources for communication and political content analysis has exponentially grown in volume and complexity. The advances in computational methods, such as supervised machine learning, allow researcher to process vast quantities of text.

Yet, communication rarely happens via a single modality in legacy and contemporary communication channels. For example, one of the key political actors in Europe, the European Union has increased its Twitter communication almost 4-folds in the last decade where more than 40% of the messages contain at lease one embedded image (Anonymized 2022). Dictionary based and shallow learning (SL) methods, unfortunately, have a hard time incorporating multi-modality into the analysis.

Deep learning (DL) brings the possiblity to extend content analysis to multi-modal materials. Previous studies have demonstrated the flexibility of embeddings to analyze multimodel data (K. Li et al., 2022; Niu et al., 2019; Tseng et al., 2021).

In this paper, we evaluate the feasibility of using multi-modal 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 multi-modal DL embedding-based models to classify manually annotated tweets from 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 multi-modal data in automated content analysis.

# Research design and data:

We use 898 tweets from the EU executive accounts sent out between December 1st 2019 and July 31st 2020. We focus on these tweets in our experiment specifically because they deliver their message by combining text and images or supplemented the textual message with imagery. Examples of them is presented in Figure 1. The tweets are manually annotated as a part of the project Ananymized (Anonymized 2022) . The orginal dataset is composed of all tweets from the 117 verified accounts of EU executives which includes commissioners, director generals, institutions and agencies responsible for policy making and implementation at the EU level. Tweets were manually coded as a whole based on whether the content of the tweet provides an update and information on political operations, policies, programs, and reports published by the EU, its institutions, or its bureaucrats. Overall there are 479 tweets that contain such message (code:1) and 419 tweets that do not (code:0). To ensure the data quality, three rounds of intercoder reliability tests were conducted between coders before coding the full sample. The first two rounds showed insufficient reliability scores, but after intensive training and discussion among coders, we reached sufficient reliability scores across all coding categories (Krippendorf’s α >.8).

Graphical user interface, text

Description automatically generated

Figure 1: Example tweets

In our experiments, we take each binary indicator as the outcome and learn individual SL and DL predictive models for them. From the SL classification family, we use Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), logistic regression, and XGBoost (XGB) while we use a Multi-Layer Perceptron (MLP) as a DL classifier. We extensively use Quanteda (Benoit et al., 2018) and caRet (Kuhn, 2008) packages in R. For text preprocessing we remove links and punctuation. Also for some SL models, we use SciKitLearn(Pedregosa et al., 2011). 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.

of e

This feature matrix is created using VisualBERT (L. H. 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 as 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 the 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.” (L. H. Li et al., 2019, p. 4)

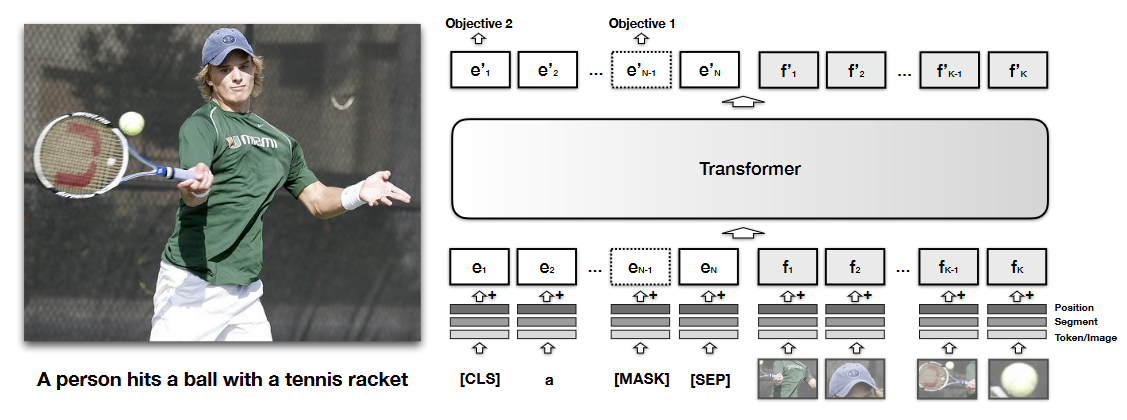


Figure 2: VisualBERT architecture, from L. H. Li et al. (2019)

of e

# Results

We report those models with hyperparameters we found to perform best. We find that including images is no sure-fire way to improve model performance. Featureization methods that are comparatively cheap when it comes to computational cost outperform multimodal embeddings as features by a large margin. Although the MLP classifier achieves to identify some useful signal in the multimodal embeddings, it is apparent that it can not make full use of them. Reasons here can be manyfold. Either because 898 samples are too few and there are tweets of a kind that only show up in the testing data. After all, MLPs are not few-shot or zero-shot learners. They need sufficient training material that resembles the testing material quite closely. Also it is possible that the image distorts the textual signal. This would be in line with findings from (HERE THE PAPER SINA WROTE).

# Discussion

These are only some first steps toward a better understanding of what a possible automated multimodal pipeline for social science purposes could look like. We aim to improve it futher

Table 1: F1 scores of predictive models

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | DFM | | | TFIDF | | | Multi-modal 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 |

# Literature

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

Loper, E., & Bird, S. (2002). *NLTK: The Natural Language Toolkit* (arXiv:cs/0205028). arXiv. https://doi.org/10.48550/arXiv.cs/0205028

Niu, Y., Lu, Z., Wen, J.-R., Xiang, T., & Chang, S.-F. (2019). Multi-Modal 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*(null), 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, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). *Detectron2*. https://github.com/facebookresearch/detectron2