ArchiMeDe @ DANKMEMES: A New Model Architecture for Meme Detection

Jinen Setpal
RN Podar School
Mumbai, India
jinens8@gmail.com
jinen.setpal@rnpodarschool.com

Gabriele Sarti
Department of Mathematics and Geosciences
University of Trieste & SISSA
Trieste, Italy
gsarti@sissa.it

Abstract

English. We introduce ArchiMeDe, a multimodal neural network-based architecture used to solve the DANKMEMES meme detections subtask at the 2020 EVALITA campaign. The system incorporates information from visual and textual sources through a multimodal neural ensemble to predict if input images and their respective metadata are memes or not. Each pre-trained neural network in the ensemble is first fine-tuned individually on the training dataset to perform domain adaptation. Learned text and visual representations are then concatenated to obtain a single multimodal embedding, and the final prediction is performed through majority voting by all networks in the ensemble.

Italiano. *Presentiamo ArchiMeDe, un'architettura multimodale basata su reti neurali per la risoluzione del subtask di "meme detection" per DANKMEMES a EVALITA 2020. Il sistema unisce informazione visiva e testuale attraverso un insieme multimodale di reti neurali per prevedere se immagini e rispettivi metadati corrispondano a meme o meno. Ogni rete neurale pre-allenata all'interno dell'insieme è inizialmente adattata al dominio specifico del dataset di training. In seguito, le rappresentazioni di ogni rete per immagini e testo vengono concatenate in un unico embedding multimodale, e la previsione finale è effettuata tramite un voto di maggioranza effettuato da tutte le reti nell'insieme.*

1 Introduction

In recent years, the democratization of data collection procedures through web scraping and crowdsourcing has led to the broad availability of public datasets spanning modalities like language and vision. Contemporary state-of-the-art machine learning models can leverage those resources to achieve highly accurate and often superhuman performances using millions or even billions of parameters (Brown et al., 2020), but are heavily reliant on an abundance of computational resources to work properly. Consequently, such architectures' training is often inaccessible to smaller research centers – let alone individual users. To counter this tendency, the availability of pre-trained open-source models has dramatically reduced the computational threshold required to obtain state-of-the-art results in multiple languages and vision tasks (Devlin et al., 2019; He et al., 2016). Pre-trained systems are often leveraged in a two-step framework: first, they undergo an unsupervised or semi-supervised pre-training to learn general knowledge representations, then they are fine-tuned in a supervised way to adapt their parameters in the context of downstream tasks. This transfer learning approach stems from the computer vision literature (He et al., 2019) but has been recently adopted for natural language processing tasks with positive results (Howard and Ruder, 2018; Devlin et al., 2019; Liu et al., 2019).

In this paper, we present ArchiMeDe, a multimodal system leveraging pre-trained language and vision models to compete in the DANKMEMES (Miliani et al., 2020) shared task at the EVALITA 2020 campaign (Basile et al., 2020). Following recent transfer learning approaches, our system leverages pre-trained visual and word embeddings in a multimodal setup, obtaining strong results on the meme detection subtask. Specifically, we participated in the first sub-

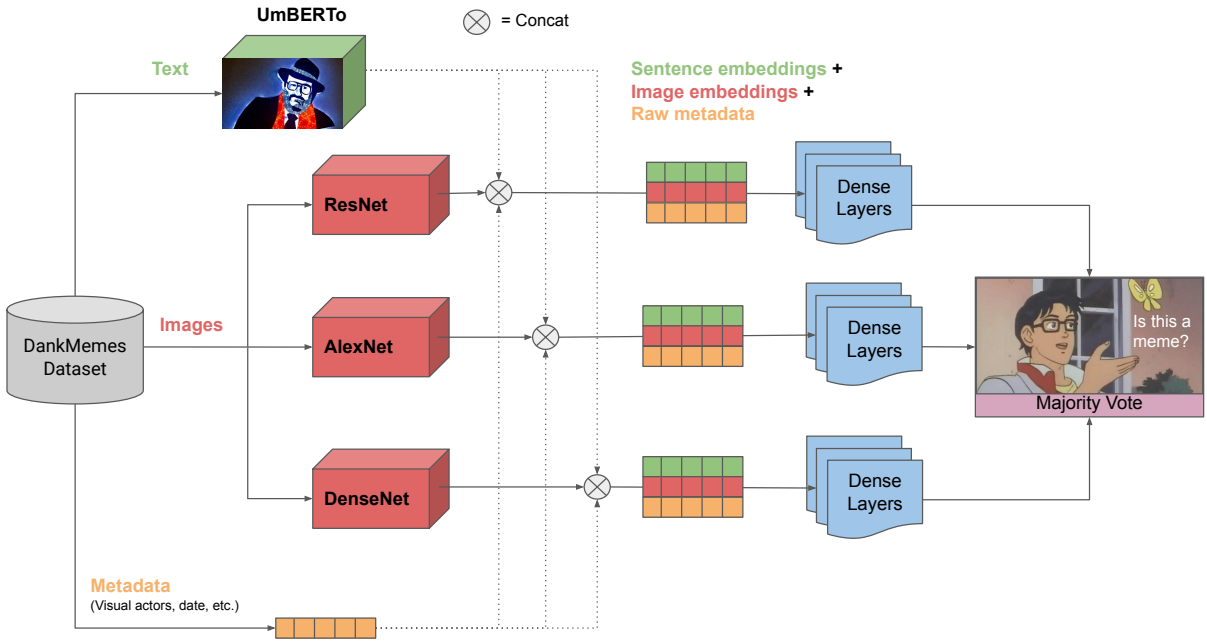

Figure 1: The ArchiMeDe system architecture. Sentence embeddings produced by the UmBERTo NLM are concatenated to metadata and image embeddings produced by three popular pre-trained vision modals. The three resulting multimodal embeddings are fed separately to feedforward networks, and the final outcome is selected through majority voting.

task of DANKMEMES, aimed at discriminating memes from standard images containing actors from the Italian political scene. Task organizers extracted a total of 1600 training images from the Instagram platform, and data available from each dataset entry – text, actors and user engagement, among others – were leveraged to train an ensemble of multimodal models performing meme detection through majority-vote. The following sections present our approach in detail, first showing our preliminary evaluation of multiple modeling approaches and then focusing on the final system's main modules and the features we leverage from the dataset. Finally, results are presented, and we conclude by discussing the problems we faced with some inconsistencies in the data. Our code is made available at `https://github.com/jinensetpal/ArchiMeDe`

2 System Description

ArchiMeDe is composed of a multimodal learning ensemble, with the final output being the result of a majority vote. Figure 1 visualizes our approach. First, the transcript associated with each image is fed to an UmBERTo (Francia et al., 2020) neural language model (NLM) pre-trained on the Italian language to produce sentence embeddings. Then, we leverage three popular pre-trained vision architectures, namely ResNet (He et al., 2016), DenseNet (Huang et al., 2017a) and AlexNet (Krizhevsky et al., 2017), to produce three independent image embeddings for each input image. These embeddings can be considered as different views over an image that may provide us with complementary information about its content. Then, each image embedding is concatenated with the sentence embedding and the raw image metadata and fed as input to an 8-layer feed-forward neural network to predict an image's meme status. The feed-forward network also includes a single dropout layer to prevent overfitting and improve generalization. Lastly, the three predictions are weighted through majority voting to obtain the final prediction of the ensemble. Other simpler strategies using a single vision model to produce image embeddings were initially envisaged as potential candidates for our submission but were finally dismissed in light of the promising performances of the ArchiMeDe ensembling approach. We discuss those perspectives in Section 4.

The remaining part of this section contains an

in-depth description of our ensemble's components, focusing on the input features that were used and how those were preprocessed to best suit learning. Moreover, we also include transfer learning specifications with some details about their impact on the overall system accuracy.

2.1 Metadata

Engagement User engagement per post is expressed as a numeric integer value. We scale and standardize engagement values to obtain a distribution centered in 0 with $\sigma = 1$. This procedure is a standard practice to avoid passing extreme absolute values as inputs for the neural network.

Date We decided to leverage temporal information in our system, building upon the intuition that memes often rely on a small set of templates that undergo a significant variation in popularity through time. Temporal information may thus provide our system with additional cues about an image's meme status in a specific time-frame. In the training dataset, dates for each post has been presented in the yyyy-mm-dd format. This date was compared with the predetermined date, 1st January 2015, to derive a numeric value representing the number of days from the date of reference. Min-max scaling is then applied to the numeric values, further deriving float numeric values between in the range [0,1], subsequently fed into each training model.

Manipulation The manipulation field provides boolean information about whether an image has been manipulated before being added to the dataset. We found this information noisy and a weak predictor of meme status; therefore, it was dropped as input.

Visual Actors Each entry was additionally provided with a list of names of the visual actors present in the frame. In the specific case of the DANKMEMES shared task, visual actors can be especially useful to identify meme images. For example, we can hypothesize that politicians who maintain a strong public presence by making claims that produce a high level of public engagement are more likely to be the subject of meme images. Moreover, some combinations of actors may be particularly likely for memes e.g. politicians belonging to parties at the political compass's antipodes. In order to produce a unified representation of visual actors for our system, we perform a one-hot encoding of all the actors occurring in the training set: if a specific politician is present in an image, the corresponding entry is true; conversely, if no such actor is present, the binary field is set to false. Actors that were not present in the training set are disregarded during evaluation: while this step is required given the context, we assume that this may significantly impact the outcome in images for which new actors were introduced.

2.2 Textual input

The analysis of textual content in meme images is critical to the success of the overall system. Indeed, ironical or satyrical comments may deeply affect the users' interpretation of an image that would otherwise be classified as normal. We note that this problem cannot be approached similarly to standard textual analytic frameworks since memes are elucidated in short, concise phrases and do not necessarily comply with standard grammatical rules. They also tend to contain slang and vernacular expressions, which, albeit conveying the intended meaning to the reader, greatly increase the need for high model capacity and ad-hoc training data. For this reason, we selected UmBERTo (Francia et al., 2020), a RoBERTa-based (Liu et al., 2019) neural language model pre-trained on Italian texts extracted from the OSCAR corpus (Ortiz Suárez et al., 2020), for producing text representations.[1] In a recent study by Miaschi et al. (2020), the model was highlighted as one of the top Italian NLMs for encoding linguistic information about social media excerpts taken from the TWITTIRÒ and PoSTWITA Twitter corpora (Cignarella et al., 2019; Sanguinetti et al., 2018). UmBERTo has a high model capability with 125M trainable parameters and was trained on online crawled data, making it suitable for processing meme language.

SentenceTransformers We use the Sentence-Transformers framework (Reimers and Gurevych, 2019) to produce sentence embeddings by averaging all word embeddings produced by the original UmBERTo model since Miaschi and Dell'Orletta (2020) showed that those are usually much more informative than the default [CLS] sentence embedding. We fine-tune representations over the available meme textual data and use them as components of our end-to-end system.

[1] `umberto-commoncrawl-cased-v1` in the HuggingFace's model hub (Wolf et al., 2019)

2.3 Visual input

While we have so far discussed only using metadata to predict our results, it is essential to address the core of a meme: the image itself. We can internally distinguish a meme from a standard image through the aforementioned broken sentence structure, meme templates, and quick and messy edits, among other aspects. As previously mentioned, memes can be very difficult to individuate when they look like standard images but gain meme status through real-world knowledge grounding.

Due to the inherently large variance in meme images' styles and contents, it is impractical to expect a single framework to effectively describe each distinguishable feature and utilize it to classify an entry. Hence, we split the representational burden across multiple pre-trained model architectures. Each of them uses a fundamentally different approach to extract image embeddings, making the resulting ensemble predictions more flexible in general settings. The three networks we used for producing image embeddings are:

ResNet Residual Networks, or ResNets (He et al., 2016), learn residual functions in relation to layer inputs. If $\mathcal{H}(x)$ is the standard underlying target mapping, ResNet layers are instead trained to fit another mapping $\mathcal{F}(x) = \mathcal{H}(x) - x$. The original mapping is thus recast into $\mathcal{F}(x)+x$. This approach makes the optimization process easier, allowing for deeper architectures. The default vector representation provided by task organizers is produced by a ResNet-50, with fifty blocks of residual layers. We use those image embeddings of size 2048 without further adjustments.

AlexNet AlexNet (Krizhevsky et al., 2017) is a vision architecture built with 5 layers of convolution and 3 fully-connected layers. AlexNet specializes in identifying depth; the network architecture effectively classifies objects such as keyboards and a large subset of animals. This fact makes AlexNet embeddings good predictors for features such as depth that are generally problematic in memes due to image subsections (e.g. text boxes). We use an embedding size of 4096 in the context of our experiments.

DenseNet Pre-trained models such as ResNet and AlexNet use a large number of hidden layers. While the increase in depth allows for better feature abstraction, it often leads to vanishing-gradient problems during training. DenseNet (Huang et al., 2017b) introduces dense blocks where the feature-maps of all preceding layers are used as inputs to the layer, and its feature-maps are used as inputs into all subsequent layers. This approach encourages feature reuse and may lead to more generalizable image embeddings. Each DenseNet image embedding has a size of 1000 weights.

The aim of using multiple vector embeddings was to cumulatively cover a significant portion of possible meme combinations and templates. As a result, is Section 4 we show how the ensemble of systems using different image embeddings leads to significant increases in validation accuracy.

	Run #	Precision	Recall	F1
Baseline		0.525	0.5147	0.5198
UniTor	1	0.839	0.8431	0.8411
	2	0.8522	**0.848**	**0.8501**
SNK	1	0.8515	0.8431	0.8473
	2	0.8317	**0.848**	0.8398
UPB	1	**0.861**	0.7892	0.8235
	2	0.8543	0.8333	0.8437
ArchiMeDe	1	0.8249	0.7157	0.7664
Keila	1	0.8121	0.6569	0.7263
	2	0.7389	0.652	0.6927

Table 1: System ranking for the DANKMEMES meme detection subtask. Top scores are in **bold**, our system is underlined.

3 Results

Table 1 presents the system ranking for the meme detection subtask. Our system placed 7th in terms of F1 score,[2] impeded primarily by inconsistent recall performances but significantly better than the random baseline (+0.2466 F1).

Results suggest that ArchiMeDe has developed inductive biases for specific image features that strongly influence the classification outcome. By inspecting validation folds over training data, we observe that most false negatives produced by the system involve distinct facial characteristics of scene actors. Inversely, ArchiMeDe effectively classifies images containing text bubbles and evident manual edits. Another notable failure case we identified is due to face-swapping. This failure is especially relevant since face-swapping is com-

[2]The F1 score is the harmonic mean between precision and recall, commonly used to evaluate classification systems.

Encoder	Precision	Recall	F1
AlexNet	.83/.77	.75/.85	.79/.81
DenseNet	.87/.83	.82/.87	.84/.85
ResNet	.83/.79	**.87/.86**	.85/.83
ResNeSt	.80/.84	.84/.76	.82/.79
ArchiMeDe	**.87/.85**	.84/.87	**.86/.86**

Table 2: Performances of ArchiMeDe variants with single image encoders over a validation split of the DANKMEMES training set. Scores are presented for non-meme/meme classes.

monly used to add an ironic component to meme images, but it is hardly detectable due to missing real-world context.

4 Other Embedding Approaches

As a complementary perspective on our experiments' nature, in this section, we present other approaches tested in the context of meme detection and that were finally disregarded in favor of the ArchiMeDe approach presented in the previous section.

CNN without Metadata Preliminary runs on the DANKMEMES dataset relied solely on the use of standard convolutional neural networks. The target architecture was fed the image itself without associated metadata to ensure that the standalone impact of the architecture was shown. The system performed poorly, performing only slightly better than the baseline scores. Additional measures to optimize this network were not taken since we assumed that this naive approach would not lead to substantial gains in performances over the baseline.

Single Pre-trained Image Encoder Before working with an ensemble, we estimated the performances of its components in performing meme detection. Besides the three models that we finally included in ArchiMeDe, we also tested ResNeSt (Zhang et al., 2020), which was finally dropped due to the similarity of its predictions to those of ResNet-50. Table 2 presents the performances of the individual image encoders and the final ensemble over a validation split containing 320 examples equally distributed over (meme, non-meme) classes. Results show how the DenseNet model appears to be better in terms of precision, while ResNet is worse but compensates with a higher recall. We found that misclassified observations were different across models, suggesting that each model could capture different properties of the input. The only exception was the ResNeSt model, which produced errors very close to the ResNet ones and was henceforth dropped for further experiments.

Multimodal Ensemble Following the complementary viewpoints of different encoders, we decided to evaluate the performances of an ensemble. Table 2 shows that our ArchiMeDe ensemble outperforms single systems in terms of both precision and recall when considering both classes, compensating the weaknesses of individual systems. The resulting majority-vote ensemble was optimized and used as the final system for our submission. Multiple experimental iterations showed that an increase in depth, followed by a reduction in layers' width, led to increased accuracy scores. Each model was trained with a batch size of 64 sets, 100 epochs fitted with test accuracy callbacks, and an early stopping strategy with a five epochs' patience value. Each model utilized the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 and was trained using a binary cross-entropy loss over the two categories.

4.1 Data Augmentation

Given the relatively small size of the available training dataset and since popular classification models are often trained using thousands if not millions of images, we tested some data augmentation strategies to improve our system's generalization performances. We applied random changes for each image to augment data, modifying it with random brightness, rotation, and zoom in a reasonable margin to keep it distinguishable. 9 augmented images were produced for every initial image entry. As a result, the training dataset is increased from 1280 to 12800 images.

Every augmented image is associated with the same metadata as the original, varying only in the visual embedding itself. The result we aimed for was an increase in generalization performances, as the model fits better to the general rule of recognizing memes. However, our results showed the opposite behavior: the system would easily overfit individual observation when data augmentation was used. We think this was partly due to augmentations not pertinent to the general meme template and partly because of the significant increase in

the number of entries having the same associated metadata.

An extensive set of augmentation strategies was tested over the dataset, modifying factors, ranges, and augmentation count. No iteration significantly and consistently improved the system's performance, and thus the augmentation process was determined noisy, relatively inconclusive, and therefore dropped from the training procedure.

5 Discussion and Conclusion

In this paper, we presented ArchiMeDe, our multimodal system used for participating in the DANKMEMES task at EVALITA 2020. The results produced by the system are promising, even if the systems do not encode inductive biases that are specific neither for multimodal artifact recognition nor to meme detection in particular. The entry is not far behind in terms of precision from the best-performing systems, and several paths display considerable potential for improving its performances. The paper effectively highlights the crucial impact of transfer learning on the success of this system. Notably, ArchiMeDe can be easily trained with standard consumer-level GPUs.

A direction that can be explored to improve the current system would be to modify the recall threshold, obtaining a better precision-recall balance for predictions. Another possibility involves introducing an aggregator network on top of the ensemble instead of using majority vote: in this way, the network can learn whether the predictions of a single subnetwork are reliable, regardless of it being part of the majority. The ensemble could also include more varied models with differing architecture to further accentuate differences in feature representations. Above all, we believe that leveraging additional data (not necessarily in Italian) could significantly improve the system's performance at the cost of increased time and computational costs.

Memes today are one of the most formidable modes of portraying one's idea while building a strong interpersonal connection between creators and users. The informality of memes, combined with their ease of making and distribution, has greatly accentuated their growth in the last few years. To be able to interpret memes effectively is a task far deeper than what can be intuitively thought. As humans continue to unravel their minds and derive ingenious computational methods, we realize the importance of slang and how it relates directly to the core human principle of community belonging. A piece of our culture, memes are the best represented and documented cultural artifacts we have today, and to effectively interpret them would mean to cross a significant milestone for the field NLP, with lasting impacts on our society as a whole.

References

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. EVALITA 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Alessandra Teresa Cignarella, Cristina Bosco, and Paolo Rosso. 2019. Presenting TWITTIRÒ-UD: An italian twitter treebank in universal dependencies. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*.

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, June. Association for Computational Linguistics.

Simone Francia, Loreto Parisi, and Magnani Paolo. 2020. UmBERTo: an italian language model trained with whole word maskings.

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

Kaiming He, Ross B. Girshick, and P. Dollár. 2019. Rethinking ImageNet pre-training. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4917–4926.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July. Association for Computational Linguistics.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. 2017a. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. 2017b. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

Alessio Miaschi and Felice Dell'Orletta. 2020. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119, Online, July. Association for Computational Linguistics.

Alessio Miaschi, Gabriele Sarti, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Italian transformers under the linguistic lens. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it)*.

Martina Miliani, Giulia Giorgi, Ilir Rama, Guido Anselmi, and Gianluca E. Lebani. 2020. DANKMEMES @ EVALITA2020: The memeing of life: memes, multimodality and politics. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online, July. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November. Association for Computational Linguistics.

Manuela Sanguinetti, Cristina Bosco, Alberto Lavelli, Alessandro Mazzei, and Fabio Tamburini. 2018. PoSTWITA-UD: an Italian Twitter Treebank in universal dependencies. In *Proceedings of the Eleventh Language Resources and Evaluation Conference (LREC 2018)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi-Li Zhang, Haibin Lin, Yu e Sun, Tong He, Jonas Mueller, R. Manmatha, M. Li, and Alex Smola. 2020. Resnest: Split-attention networks. *ArXiv*, abs/2004.08955.