Team art-nat-HHU at SemEval-2024 Task 8: Stylistically Informed Fusion Model for MGT-Detection

Vittorio Ciccarelli[†]
Heinrich-Heine-Universität
Düsseldorf (HHU)
vittorio.ciccarelli@hhu.de

Cornelia Genz[†]
Heinrich-Heine-Universität
Düsseldorf (HHU)
cornelia.genz@hhu.de

Nele Mastracchio[†]
Heinrich-Heine-Universität
Düsseldorf (HHU)
nele.mastracchio@hhu.de

Wiebke Petersen[†]
Heinrich-Heine-Universität
Düsseldorf (HHU)
wiebke.petersen@hhu.de

Anna Sophia Stein[†]
Heinrich-Heine-Universität
Düsseldorf (HHU)
anna.stein@hhu.de

Hanxin Xia[†]
Heinrich-Heine-Universität
Düsseldorf (HHU)
hanxin.xia@hhu.de

Abstract

This paper presents our solution for subtask A of shared task 8 of SemEval 2024 for classifying human- and machine-written texts in English across multiple domains. We propose a fusion model consisting of a RoBERTa-based pre-classifier and two MLPs that have been trained to correct the pre-classifier using linguistic features. Our model achieved an accuracy of 85%.

1 Introduction

After rapid developments in large language models (LLMs) and generative AIs in the last years, the detection of machine-generated content has become one focus of study as deepfakes, machine-generated lawyer statements and even libel suits (Superior Court of Gwinnett County) concerning language machines stress the importance of detecting texts not written by humans. The SemEval shared task 8 in 2024 aims at multi- and monolingual machine-generated text (MGT) detection from various domains by multiple models.

For the monolingual English data in subtask A (Wang et al., 2024) we propose a fusion model built using pre-trained RoBERTa word embeddings specialized for AI-generated text detection and correction MLP classifiers, supported by the additional computation of linguistic, stylistic and probabilistic features selected based on their informational value. With this system design, our model ranked at position 25 out of 124 with an 0.855 accuracy score on the task. The only data used for training was the one provided by the organizers without further data augmentation. Because of the different distributions of the data in the development and test data sets several strategies were tested and a fusion model was chosen as the best strategy. The fine-tuned RoBERTa Base OpenAI Detector alone performed well but developed a bias towards the machine class. To stabilize the model linguistic, probabilistic and stylistic features were added, which improved the overall F1 score of the fusion architecture.

2 Background

Over the last years, numerous approaches have been proposed to tackle the task of MGT detection. Some models, such as DetectMGT (Mitchell et al., 2023), focus on detecting texts from a specific source, such as GPT-family LLMs, while other approaches are specialized in texts from a specific genre, such as Shijaku and Canhasi (2023) for TOEFL essays. Other architectures, like ensemble models combining different classifiers (del Campo-Ávila et al., 2007) have been successfully used for machine-generated text detection to improve out-of-distribution performance (Lai et al., 2024).

Guo et al. (2023) show that, overall, deep-learning approaches, and in particular a RoBERTa-based-detector, are one of the best individual models for MGT detection. The RoBERTa-based-detector was shown to be particularly robust against oov scenarios in both Chinese and English, compared to a machine learning model. Moreover, Wang et al. (2023) and He et al. (2024) conducted large-scale bench-marking on existing approaches for MGT detection across multiple domains, models, and languages and concluded that the RoBERTa language model, especially the variants that have been optimized for AI detection tasks, consistently outperforms most other methods across evaluation metrics.

[†] Equal contribution.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1690–1697
June 20-21, 2024 ©2024 Association for Computational Linguistics

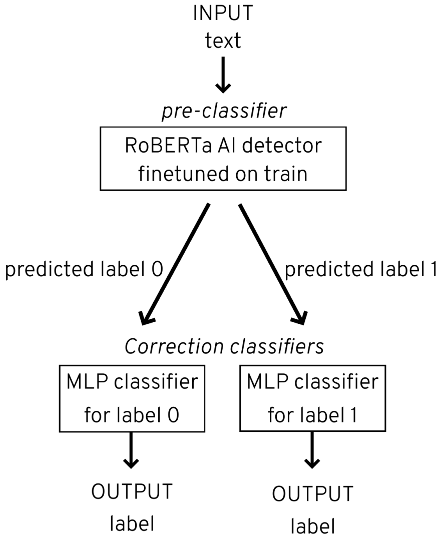

Figure 1: Model architecture used to obtain submission results.

3 System overview

Our system is based on a pre-classifier that is a RoBERTa model fine-tuned for AI generated text detection. In order to correct the predictions of the pre-classifier, two correction classifiers have been trained that are based on linguistic, stylistic and probabilistic features. An overview of the system setup is given in Figure 1.

Our RoBERTa pre-classifier is based on the RoBERTa Base OpenAI Detector[1] (Solaiman et al., 2019), a RoBERTa model fine-tuned for AI generated text detection. This model has been further fine-tuned on 10% of the training data. A prediction was then generated for each text in the train, dev and test set. In order to improve the predictions, two correction Multi-layer Perceptron (MLP) classifiers, one for each label, were trained on the training and development data of their respective label (see Figure 1) as well as on a range of features outlined in Section 4.2.1. To generate the final classification, all texts were classified again by the correction MLP that corresponded to the label predicted by the RoBERTa pre-classifier. This provided an opportunity for the more specialized classifier to correct the initial prediction.

4 Experimental setup

The M4 dataset consists of both machine (label 1) and human-generated texts (label 0). The dataset features texts from six different LLM generators (Davinci, chatGPT, Dolly, Cohere, BLOOMz and GPT4) and five different genres (Reddit, WikiHow, ArXive, Wikipedia, and peerRead). Participants were provided first with a train and dev set and later with a test set with 119,757, 5,000, and 34,272 texts in total, respectively.

Roughly 53% of the documents in the train set are machine generated (DaVinci: 14,343, chatGPT: 14,339, Dolly: 14,046, Cohere: 13,678), and 47% are human-written. In the dev set, exactly half of the texts were machine-generated by the BLOOMz model, the other half was human-written. The test set contains 18,000 (53%) machine-generated texts from Davinci, chatGPT, Dolly, Cohere, BLOOMz and GPT4 (3,000 texts each) and 16,272 (47%) human-written texts. An overview of the data is provided in Table 1. Since we did not include genre- or machine-specific information for our approach, this information is excluded from the table.

	train	dev	test
machine	53%	50%	53%
human	47%	50%	47%
total texts	119,757	5,000	34,272

Table 1: Label distribution across train, dev and test set.

4.1 RoBERTa pre-classifier

We used a fine-tuned RoBERTa Base OpenAI Detector as our pre-classifier. Because the OpenAI Detector had already been fine-tuned for human-machine classification, and to facilitate replication of the experiment, we used only 10% of the training data to further fine-tune the model[2].Training was done for 3 epochs with a learning rate of $2e^{-5}$ on Google Colab using a T4 run-time and took 45 minutes.

4.2 Correction classifiers

4.2.1 Feature extraction

To capture characteristics of machine-generated and human-written texts, the data was analyzed for various linguistic features. Altogether 70 features, widely used in NLP and easy to compute, were extracted, 35 of which exhibited a high to medium correlation with the gold label (see Table 4 in the Appendix). All features were computed on a 24GB

[1] https://huggingface.co/openai-community/roberta-base-openai-detector

[2] The data for fine-tuning consisted of 2,000 texts of each author category (Davinci, Cohere, Dolly, chatGPT, and human).

RAM machine with a Ryzen 7 7730U, which took up to 6 hours for all texts depending on the feature.

Count-based features. The texts were split into words and punctuation using regular expressions to derive the following features: mean sentence length, ratio of punctuation to words, ratio of word types to tokens, ratio of vowels to words and mean word length. The NLTK stopword list was used to get the ratio of content words to other words. Additionally the number of hapax legomena per text and the number of negation words (manually compiled list) per text were computed.

Syntactic features. All texts were POS-tagged with the NLTK part of speech tagger to compute syntactic features: ratio of nouns, verbs, adjectives, adpositions, adverbs, conjunctions, numerals, pronouns and determiners to words alltogether, ratio of adjectives to nouns and ratio of verbs to nouns.

Using the dependency parser of Spacy (Honnibal and Montani, 2017) we extracted the maximum depth of a dependency tree, mean depth of all dependency trees in a text, and number of passive constructions (determined by the number of *nsubj-pass* POS tags) per sentence.

Frequency features. To capture whether the texts differ in word use, the logarithmic frequency of all content words in the human texts were computed. Additionally, lists of frequent words (frequency $\geq$ 12) and hapax legomena (frequency = 1) have been computed. From this the following features were extracted for all texts: mean log frequency of content words, ratio of frequent words to content words, ratio of hapax legomena to content words and number of hapax legomena.

Additionally, we used the Wiktionary frequency lists for English[3] and extracted a list of high frequency words (top 10%), mid-high frequency words (top 20%) as well as field specific word lists, namely the most frequent words in fantasy texts and in Wikipedia articles. For each list and each text in the datasets we extracted the ratio of words belonging to the lists to the content words as a feature.

Word difficulty features. The CEFR-J[4] project provides vocabulary lists for the different proficiency levels of the Common European Framework of Reference for Languages (CEFR)[5]. We used these lists[6] to compute the following features: ratio of A1/A2/B1/B2/C1/C2-level words to content words. This was done twice: once on the basis of the stemmed and once on the lemmatized words. We used the Porter stemmer and the WordNet lemmatizer from NLTK.

Stylistic and sentiment features. A number of features concerning text style and text sentiment were extracted. Using the same method as for the difficulty features above, we extracted the ratio of words in the list of negative opinion words compiled by Liu et al. (2005) as well as the readability score of the texts according to the Flesch reading-ease test[7] . The other features in this subset have been extracted by using available fine-tuned classifiers. Emotion English DistilRoBERTa-base[8] is a classifier that predicts Ekman's six basic emotions, plus a neutral class (cf. Hartmann, 2022). The logit for each class provides one feature (anger, disgust, fear, joy, neutral, sadness, surprise). As a standard sentiment analyzer we used the sentiment-analysis-pipeline from Hugging Face[9] and, using the logits, extracted two features (positive, negative). Analogously, the features 'formal' and 'informal' were extracted using the formality ranker by Babakov et al. (2023), which is a RoBERTa model trained to predict to which register a sentence belongs. Finally, we used a toxicity classification model[10] that is a RoBERTa model fine-tuned to predict whether a text is toxic or not.

Features extracted from the pre-classifier. In order to inform the correction classifiers on the basis of the decision of the pre-classifier, we extracted the logits and the last hidden state of our RoBERTa pre-classifier for each text. The last hidden states were reduced from 768 to 2 dimensions using PCA (principal component analysis) and UMAP (uniform manifold approximation and projection). For UMAP the hyperparameters `min_dist`, `n-neighbors` and `metric` were tuned by a combination of random search and grid search

[3] https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/English

[4] https://www.cefr-j.org/

[5] https://www.coe.int/en/web/common-european-framework-reference-languages

[6] https://github.com/openlanguageprofiles/olp-en-cefrj/tree/master

[7] https://github.com/textstat/textstat

[8] https://huggingface.co/j-hartmann/emotion-english-distilroberta-base

[9] https://huggingface.co/

[10] https://huggingface.co/s-nlp/roberta_toxicity_classifier

and evaluated on the accuracy of a logistic regression classifier that predicts the label from the 2 dimensions. The extracted features included in our feature set are the logits, the 2-dimensional PCA representation of the last hidden state and the 2 UMAP dimensions gained by setting `min_dist` to 0.01 and `n-neighbors` to 100. We kept two metrics, namely `cosine` and `jaccard`.

4.2.2 Feature selection

To account for the variability in features, we initially scaled all 70 features using the Standard Scaler from scikit-learn (Pedregosa et al., 2011). During the collection of the 70 features, no attention was paid to whether they contained quasi-duplications. Features which were highly correlated with other features (>0.9) were removed subsequently using a correlation matrix. After this removal, 51 features remained.

In the next step, only features with high or medium correlation with the gold label (Pearson correlation ≥ 0.1 or ≤ -0.1) were retained in order to choose the features most relevant for the classification task. Table 4 in the Appendix shows all features (including those which are highly correlated to each other) that have at least a medium positive or negative correlation with the gold label. After both selection steps, 26 features remained (see Table 5).

4.2.3 Model selection and training

In a comparison of various classifiers from scikit-learn (i.e. Random Forest, Logistic regression), MLPs performed best in most settings: whether trained on all features, trained only on at least medium correlated features, or trained only on features that are not extracted from the pre-classifier. We therefore chose MLP as our correction classifiers.

Before conducting training on the combined train and dev dataset, we separated the texts for which the RoBERTa pre-classifier had predicted the human label from those for which it had predicted the machine label, thus creating two splits. Then, we trained two separate MLPs on the two splits of the training data using the 26 features identified as relevant in the feature selection process (4.2.2). The idea behind this approach was that the models might learn in which cases the fine-tuned transformer classified the data incorrectly, and would thus have to be corrected. The test data was then prepared by calculating the 26 features, on which the pair of MLP correction classifiers made the final predictions.

model	label	prec.	rec.	f1
fusion model	human	0.85	0.85	**0.85**
	machine	0.86	0.86	**0.86**
accuracy: **0.85**				
pre-classifier	human	**0.99**	0.48	0.64
	machine	0.68	**0.99**	0.81
accuracy: 0.75				
MLP	human	0.53	0.89	0.67
	machine	0.75	0.30	0.43
accuracy: 0.58				

Table 2: Precision, recall, f1-score, and overall accuracy for the submitted fusion model and two models for comparison: the RoBERTa pre-classifier and an MLP model trained with the selected linguistic features. The support for the ‘human’ class is 16,272 and for the ‘machine class 18,000.

5 Results

Table 2 shows the performance of the submitted fusion model, obtained using the `classification_report` from scikit-learn. Overall, the fusion model achieves an accuracy of 85%. The table additionally shows the performances of two other models on the test data in comparison: (i) the RoBERTa pre-classifier; (ii) an MLP model that was trained with the same hyperparameters as used for the correction classifiers and the same features selected in the feature selection process (see Section 4.2.2), except for the ones extracted from the pre-classifier (see Section 4.2.1).

Although the pre-classifier performed fairly well on the dev data (accuracy: 0.89, for more details see Table 6 in the Appendix), we opted for a fusion model with a correction layer in order to improve robustness for data from new generators and domains. The implementation of the two correction MLPs corroborated the hypothethis. On the test data, the accuracy of the pre-classifier drops to 0.75, while the addition of the correction layer improved the accuracy to 0.85.

A closer look at the recall and precision for the two classes ‘human’ and ‘machine’ reveals that the fusion model balances out the problems of the pre-classifier and the MLP. The high precision and low recall of the pre-classifier for ‘human’ and vice versa for ‘machine’ indicate that it is biased towards the ‘machine’ class. Accordingly the relatively high precision and low recall of the MLP classifier for ‘machine’ and vice versa for ‘human’ indicate that it is biased towards the ‘human’ class. In contrast, the fusion model shows equally high precision and recall for both classes.

As described in Section 4.1, the pre-classifier is obtained from the RoBERTa base OpenAI detector by further training. Comparing its performance to the original model (see Table 7, Appendix), fine-tuning has led from a bias towards the ‘human’ class to a bias towards the ‘machine class’. This is likely due to the fact that the fine-tuning data had an imbalance towards machine-generated texts.

5.1 Error Analysis

An error analysis was completed in three parts: We examined the influence of the different labels, the features, and the correctional classifiers on accuracy. The influence of the domain was not examined since there was only one domain present in the test data.

When inspecting the label distribution for the misclassified texts, we can see an almost perfect 50% split between human and machine-labeled texts. Between the models, the errors are not distributed as evenly, as shown in Figure 3 in the Appendix. GPT4 and dolly texts were misclassified most often, followed by Cohere, DaVinci, and chatGPT, while BLOOMz texts were rarely classified incorrectly. Since GPT4 texts were not seen in the train or dev data, it is not unsurprising that those texts were classified least accurately. A further reason could be that GPT4-generated texts are known to be very ‘human-like’, hence harder to differentiate from human texts.

Figure 2 shows the classification by the pre-classifier and whether it was modified by the correction classifier (the same data in numbers is given in Table 3). A text identified as human by the pre-classifier was typically classified accurately and only rarely adjusted by the correcting classifier. For the texts where the pre-classifier predicted a machine label, the prediction was corrected often. However, as shown in table 3, 2,308 cases should have been corrected and were not. The machine label predictions by the pre-classifier have caused most errors, as that label was predicted so often. This is also reflected in the recall of the pre-classifier-only model in Table 2.

Finally, we correlated all features used in the fusion model with the labels that were predicted incorrectly. The strongest correlation was shown by the features *1st UMAP-dimension (Jaccard)* (-0.82), *ratio of CEFR-B1 words (stem)* (- 0.71), *ratio of CEFR-B2 words (stem)* (- 0.57), *neutral sentiment score* (- 0.51), and *ratio of pronouns to content words* (0.55). Since the correlations are on the wrong predictions, a strong negative correlation indicates a correlation with an incorrectly predicted human label (label 0), while a strong positive correlation implies the opposite. It is possible that the low correlation threshold chosen for feature acceptance led to the inclusion of features initially weakly correlated with the labels in the training and development data, which may have adversely affected the correctional classifiers’ decisions. Alternatively, the test set data might exhibit a different distribution for those features compared to the training and development data.

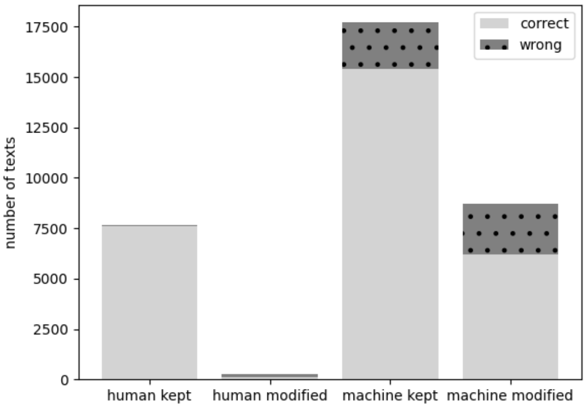

Figure 2: Left two bars: predictions by correction classifier correcting texts pre-classified as “human”, right two bars: predictions by correction classifier correcting texts pre-classified as “machine”.

pre-classifier	h		m	
correction classifer	h	m	m	h
correct	7622	86	15402	6189
wrong	12	153	2308	2500

Table 3: Classification errors split by prediction by the pre-classifier and correction classifier (h = human, m = machine).

6 Conclusion and Limitations

Overall, this study has highlighted the benefit of using a fusion architecture consisting of a pre-classifier and linguistically informed correctional classifiers. By adding syntactic, stylistic, sentiment, frequency- and word difficulty-based features, we were able to improve the performance of a fine-tuned pre-trained RoBERTa model for AI generated text detection and adjust the bias towards the machine label. Because our fusion model uses a pre-trained RoBERTa model, all computations for this paper can be run locally or, in the case of the RoBERTa fine-tuning, using a free Google Colab account. This means that our model can be easily expanded and leaves a smaller environmental footprint.

Future studies could expand our fusion model by incorporating more semantic-level or complex features such as contextual predictability, as well as fine-tuning the pre-classifier using more, balanced data. Our code is available on GitHub[11].

References

Nikolay Babakov, David Dale, Ilya Gusev, Irina Krotova, and Alexander Panchenko. 2023. Don't lose the message while paraphrasing: A study on content preserving style transfer. In *Natural Language Processing and Information Systems*, pages 47–61, Cham. Springer Nature Switzerland.

José del Campo-Ávila, Gonzalo Ramos-Jiménez, and Rafael Morales-Bueno. 2007. Incremental learning with multiple classifier systems using correction filters for classification. In *Advances in Intelligent Data Analysis VII*, pages 106–117, Berlin, Heidelberg. Springer Berlin Heidelberg.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Jochen Hartmann. 2022. Emotion english distilroberta-base. `https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/`.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. 2024. Adaptive ensembles of fine-tuned transformers for llm-generated text detection. *arXiv preprint arXiv:2403.13335*.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Superior Court of Gwinnett County. Mark Walters v. OpenAI,L.L.C.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

[11] `https://github.com/ansost/art-nat-HHU-semeval2024`

A Features with medium or high correlation

feature	corr.
1st UMAP-dimension (jaccard)	0.94
logits for label 1 from pre-classifier	0.92
roBERTa prediction	0.92
positive sentiment score	0.28
ratio of determiners to content words	0.28
ratio of pronouns to content words	0.25
score for formal	0.22
ratio of CEFR-B1 words (stem)	0.20
ratio CEFR (all levels) words (stem)	0.17
ratio of CEFR-B2 words (stem)	0.17
ratio of CEFR-A2 words (stem)	0.13
ratio of CEFR-B1 words (lemma)	0.13
ratio of conjunctions to words	0.13
ratio of CEFR-A2 words (lemma)	0.12
score for joy	0.11
ratio of fantasy words	0.10
score for neutral	0.10
ratio of Wikipedia words	0.10
word ratio of top 10% freq. Wiktionary words	0.10
word ratio of top 20% freq. Wiktionary words	0.10
. . .	
score for fear	−0.10
1st UMAP-dimension (cosine)	−0.10
score for anger	−0.11
ratio of pronouns to words	−0.15
number of hapaxes	−0.17
score for informal	−0.22
ratio of adverbs to words	−0.22
prop. of unfreq. words to content words	−0.25
TTR	−0.27
number of unique words	−0.27
negative sentiment score	−0.28
mean depth of dep. tree for sentences	−0.40
max depth of dependency tree	−0.45
2nd UMAP-dimension (jaccard)	−0.59
logits for label 0 from pre-classifier	−0.92

Table 4: Features with medium or strong positive or negative correlation ($-0.1 \leq \text{corr} \leq 0.1$) with label 1 (machine) in train data

B Fusion model features

feature name
type-to-token ratio (TTR)
ratio of adverbs to content words
ratio of pronouns to content words
ratio of determiners to content words
ratio of conjunctions to content words
ratio CEFR (all levels) words
ratio of CEFR-A2 words
ratio of CEFR-B1 words
ratio of CEFR-B2 words
number of hapaxes
ratio of frequent words to content words
ratio of hapaxes to content words
1st UMAP-dimension (cosine)
negative sentiment score
positive sentiment score
score for anger
score for fear
score for neutral
score for joy
score for formal
score for informal
max depth of dependency tree
mean depth of dependency tree for sentences
word ratio of top 10% freq. Wiktionary words
1st UMAP-dimension (jaccard)
logits for label 0 from RoBERTa pre-classifier

Table 5: Features used to train the fusion model.

C RoBERTa pre-classifier performance on dev

label	precision	recall	f1-score
human	0.91	0.86	0.88
machine	0.87	0.91	0.89

Table 6: Precision, recall, f1-score, and support for the RoBERTa pre-classifier on the dev data. Accuracy is 0.89

D RoBERTa base OpenAI detector performance on test

label	precision	recall	f1-score
human	0.57	**0.98**	0.72
machine	**0.95**	0.34	0.50

Table 7: Precision, recall, f1-score, and support for the RoBERTa base OpenAI detector on the test data. Accuracy is 0.64

E Distribution of errors

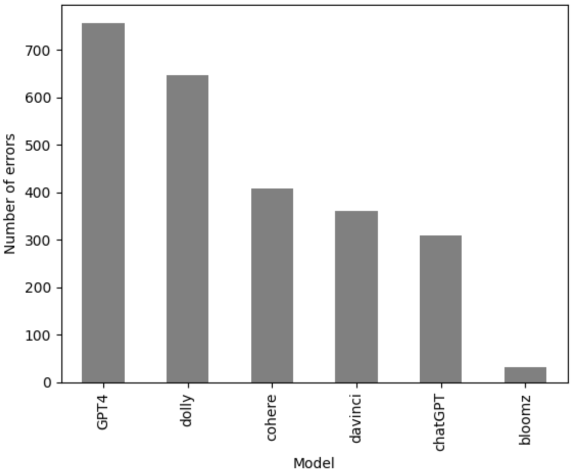

Figure 3: Distribution of models in false predictions.