Submitted 02/2024; Published 03/2024

The CLIN33 Shared Task on the Detection of Text Generated by Large Language Models

Pieter Fivez[*] PIETER.FIVEZ@UANTWERPEN.BE
Walter Daelemans[*] WALTER.DAELEMANS@UANTWERPEN.BE
Tim Van de Cruys[♠] TIM.VANDECRUYS@KULEUVEN.BE
Yury Kashnitsky[♥] Y.KASHNITSKIY@ELSEVIER.COM
Savvas Chamezopoulos[♥] S.CHAMEZOPOULOS@ELSEVIER.COM
Hadi Mohammadi[♦] H.MOHAMMADI@UU.NL
Anastasia Giachanou[♦] A.GIACHANOU@UU.NL
Ayoub Bagheri[♦] A.BAGHERI@UU.NL
Wessel Poelman[◇♠] WESSEL.POELMAN@KULEUVEN.BE
Juraj Vladika[◇] JURAJ.VLADIKA@TUM.DE
Esther Ploeger[♣] ESPL@CS.AAU.DK
Johannes Bjerva[♣] JBJERVA@CS.AAU.DK
Florian Matthes[◇] MATTHES@TUM.DE
Hans van Halteren[♡] HANS.VANHALTEREN@RU.NL

[*] *UAntwerpen, Antwerp, Belgium*
[♠] *KU Leuven, Leuven, Belgium*
[♥] *Elsevier, Amsterdam, the Netherlands*
[♦] *Utrecht University, Utrecht, the Netherlands*
[◇] *Technical University of Munich, Munich, Germany*
[♣] *Aalborg University, Copenhagen, Denmark*
[♡] *Radboud University, Nijmegen, the Netherlands*

Abstract

The Shared Task for CLIN33 focuses on a relatively novel yet societally relevant task: the detection of text generated by Large Language Models (LLMs). We frame this detection task as a binary classification problem (LLM-generated or not), using test data from up to 6 different domains and text genres for both Dutch and English. Part of this test data was held out entirely from the contestants, including a ”mystery genre” which belonged to an unknown domain (later revealed to be columns). Four teams submitted 11 runs with substantially different models and features. This paper gives an overview of our task setup and contains the evaluation and detailed descriptions of the participating systems. Notably, included in the winning systems are both deep learning models as well as more traditional machine learning models leveraging task-specific feature engineering.

1. Introduction

The widespread availability and quality of Large Language Models (LLMs) like ChatGPT confronts many organizations (teaching institutions, social media moderators, publishers, etc.) with an urgent problem: decide who has written an assignment, tweet, paper or other document: human or machine. Currently, no accurate and reliable tools exist for English, let alone other languages, despite largely unsubstantiated claims.

The goal of our shared task, organized in Antwerp in the framework of the 2023 Computational Linguistics in the Netherlands Conference (CLIN33), was to create a *test* dataset[1] representing a

[1] The generated test data is published with the DOI **10.5281/zenodo.10732797** and is available at `https://zenodo.org/records/10732813`. The not-generated (human-written) texts were made available to the participants but cannot be distributed for copyright reasons.

	CLIN33[5]	AuTexTification[6]	COLING 2022[7]	Kaggle 2023[8]
Binary detection	✓	✓	✓	✓
Training data provided	×	✓	✓	✓
Extra training data allowed	✓	×	✓	✓
Cross-domain data	✓	✓	×	×
Domain transfer allowed	✓	×	✓	✓
Multilingual data	✓	✓	×	×
Includes zero-shot test	✓	✓	×	×
Awards prize money	✓	×	×	✓

Table 1: Contrasting recent LLM detection competitions according to their task setup.

realistic context for detection algorithms: different genres, different levels of adversarial prompting, different LLMs, and multilingual (English and Dutch). The consequence of this setup is that the participants had only access to a sample of the test data during development, not to training data. This way, we simulate a realistic problem setting for useful detection systems, as generated texts do not come labeled with model, genre, and prompting information.

Our shared task can be contextualized within an international research effort toward the detection of natural language free-text generated with LLMs. This research is currently still hindered by a lack of peer review and the instability of the rapidly changing ecosystem of LLMs. The organization of detection competitions similar to our shared task acts as a main impetus for systematic evaluation and comparison of proposed models that attempts to keep up with this rapid change.

Table 1 contrasts recent detection competitions along various dimensions of the defined LLM detection task. Other subtasks than binary LLM detection also occur in such competitions: both the AuTexTification[2] task and the Machine Learning Model Attribution Challenge[3] (Merkhofer et al. 2023) require participants to attribute LLM-generated text to a specific model, while the Trojan Detection Challenge[4] is aimed at detecting hidden functionality within LLMs. The COLING 2022 competition (Kashnitsky et al. 2022) has shown that while it might be easy to achieve good detection accuracy with a specific, even large dataset, still generalization to similar datasets can be very challenging. One of the winners of COLING 2022 demonstrated (Rosati 2022) that the models trained with the competition data fail to generalize to a very similar synthetic dataset.

The insights from these competitions are complemented by research literature that focuses on crucial challenges associated with the LLM detection task, such as paraphrasing attacks (Sadasivan et al. 2023), robustness to cross-domain and multilingual settings (Antoun et al. 2023, Rosati 2022), as well as the adversarial impact of advanced LLM prompting to evade detection (Lu et al. 2023). Other challenges include the interpretability of detection models, robustness to targeted adversarial attacks, and human-machine collaboration (Jawahar et al. 2020). In this paper we focus on the shared task competitions and do not try to be complete in summarizing the related research.

[2] https://sites.google.com/view/autextification/

[3] https://mlmac.io

[4] https://trojandetection.ai

[5] https://sites.google.com/view/shared-task-clin33/

[6] https://sites.google.com/view/autextification/

[7] https://www.kaggle.com/competitions/detecting-generated-scientific-papers

[8] https://www.kaggle.com/competitions/llm-detect-ai-generated-text

Table 2: Data distribution for the Dutch detection task

	Human-generated	LLM-generated	Total
News	200	200	400
Twitter	200	200	400
Reviews	200	200	400
Poetry	50	50	100
Column	100	100	200

Table 3: Data distribution for the English detection task

	Human-generated	LLM-generated	Total
News	200	200	400
Twitter	200	200	400
Reviews	200	200	400
Poetry	50	50	100
Column	40	40	80
Open-source	25	25	50

2. Data and Systems

2.1 Data Creation Process

2.1.1 Text Genres

We have included 5 main text genres in our test data:

1. Medium-length news articles
2. Tweets from the social media platform X (previously known as Twitter)
3. Product reviews
4. Short-form poetry
5. Columns

An additional sixth genre, Open-source, was created by using an open-source LLM for generating English news data. Tables 2 and 3 give an overview of the data distribution of the entire Dutch and English test data, including the human-reference data as well as the LLM-generated data.

2.1.2 Human-reference Data

News articles We selected texts from two sources: Dutch articles from De Standaard[9] and English articles from The New York Times[10]. The articles from these sources were classified into one of 17 top-level IPTC media topics[11], employing a distance-based method with contextual embeddings (Kosar et al. 2023). To obtain balanced data, we randomly sampled data across three topics: “Politics”, “Economy, Business and Finance”, and “Science and Technology”, selecting articles from each year in the range of 2000 to 2007.

X (formerly Twitter) We used Dutch tweets about COVID-19 vaccine stance collected within the Vaccinpraat project (Lemmens et al. 2021). For English, we scraped relevant tweets based on specific keywords and hashtags related to 3 topics: MeToo, the FIFA World Cup, and COVID-19.

[9] `https://www.standaard.be`

[10] `https://www.nytimes.com`

[11] `https://www.iptc.org/std/NewsCodes/treeview/mediatopic/mediatopic-en-GB.html`

Product reviews To enforce consistency between the texts, all selected product reviews are between 200 and 400 words long. For Dutch, we used book reviews from the Dutch Book Reviews Dataset [12]. For English, we used heterogeneous product reviews from the Multilingual Amazon Reviews Corpus (Keung et al. 2020).

Poetry The Dutch poems were selected from Poëzie-Leestafel[13], while the English poems were selected from Poetry Foundation[14].

Columns Columns were the 'mystery genre' in our shared task; no information was provided beforehand about the genre. As human-generated text, we took a selection of published columns of the Flemish author Koen Meulenaere (writing under the pseudonym Kaaiman) for Dutch and of the UK journalist Laura Kuenssberg for English.

2.1.3 LLM-Generated Data

Figure 1: An excerpt of a generated English news article.

> In an unexpected turn of events, Russia's state-controlled gas company, Gazprom, today withdrew from a previously agreed deal with media mogul, Vladimir V. Gusinsky. Gusinsky, the owner of NTV, Russia's largest independent television station, has found himself in the crosshairs of this latest development. The announcement comes as federal prosecutors continue to push for his arrest over alleged financial misconduct.
>
> The agreement, which was initially meant to provide a lifeline for the beleaguered media tycoon, involved Gazprom purchasing a significant stake in Gusinsky's Media-Most holding company. This move was seen as a way for Gusinsky to protect his media empire from the Kremlin's increasing encroachment. However, Gazprom's abrupt withdrawal from the agreement has now left Gusinsky's future and that of his media empire hanging in the balance.
>
> The state-controlled gas company has yet to provide a detailed explanation for its sudden change of heart. However, insiders suggest that Gazprom's decision may be linked to the growing legal troubles facing Gusinsky. The media baron is currently under investigation by federal prosecutors over allegations of fraud and embezzlement, charges that he vehemently denies. [...]
>
> The unfolding saga between Gazinsky, Gazprom, and the Kremlin paints a picture of a complex and increasingly tense relationship between the Russian government and its media. It is a situation that observers will be watching closely, as it could have far-reaching implications for media freedom in Russia. The outcome could very well determine the future of independent journalism in the country.

We use 3 different LLMs to generate data: the closed-source ChatGPT[15] and GPT-4 (OpenAI et al. 2023) models as well as the open-source Vicuña-13B model[16] (Chiang et al. 2023). We list all used prompts in Appendix A. The prompts for the news articles, X tweets, and product reviews were designed by Pieter Fivez; those for poetry by Tim Van de Cruys; and those for columns by Walter Daelemans. As a result, these prompts not only differ in authorship, but also in their adversarial

[12] `https://github.com/benjaminvdb/DBRD`

[13] `http://www.poezie-leestafel.info/`

[14] `https://www.poetryfoundation.org/`

[15] `https://chat.openai.com/`

[16] `https://lmsys.org/blog/2023-03-30-vicuna/`

nature: this simulates a real-world scenario where LLM-generated texts from different domains can have different prompt authors, which could additionally impact the transferability of detection models.

News articles Figure 1 shows an excerpt of a generated English news article. We used both ChatGPT and GPT-4 to generate 100 news articles each for both the Dutch and English data. For the English Open-source data, we generated 25 news articles using the Vicuña-13B model. We used the exact same prompting template for all LLMs, which utilizes the headlines from our human-reference data.

Figure 2: Examples of two English and two Dutch generated tweets.

> Expressing gratitude for the incredible support and solidarity during these challenging times. Together, we can overcome the barriers and create a safer world for all. Let’s continue to raise our voices and stand against #SexualAssult. #Thankful for the progress made, but still a long way to go. #MeToo #16Days #Justice
>
> Hey @elonmusk, I’ve been a loyal Tesla owner for years but my Model 3 has been experiencing recurring issues. As a responsible driver, safety is my top priority. Can you please address this matter urgently? #VehicleSafetyMatters
>
> Beste Nederlanders, ik waarschuw jullie, deze #vaccins zijn gevaarlijk en nog in de #experimentele fase, net als die andere. Denk na en neem geen risico met je gezondheid. #vaccinatiedwang
>
> Te veel Nederlanders staan onder druk om zich te laten vaccineren, terwijl de bijwerkingen en lange termijn effecten onduidelijk zijn. Waarom wordt er niet meer gesproken over natuurlijke immuniteit en preventieve gezondheidszorg? #vaccindwang #vrijheid #artsencollectief”

X Figure 2 shows examples of generated Dutch and English tweets. We used both ChatGPT and GPT-4 to generate 100 tweets each for both the Dutch and English data. We used the exact same prompting template for all LLMs, which utilizes hashtags and style examples from our human-reference data.

Product reviews Figure 3 shows an example of a generated Amazon product review. We used both ChatGPT and GPT-4 to generate 100 tweets each for both the Dutch and English data. We used the exact same prompting template for all LLMs, which utilizes hashtags and style examples from our human-reference data.

Poetry Figure 4 shows an example of a generated Dutch poem. ChatGPT and Vicuña-13B were used to generate 50 poems for both Dutch and English.

Columns Figure 5 shows an example of a generated Dutch column. For Dutch, we generated columns in the style of Koen Meulenaere and Laura Kuenssberg by using in-context learning by adding 4-7 randomly chosen real columns (a different random selection each time) to the prompt as examples of the style. Leads from newspaper articles that could be found online on the websites of De Standaard (a Flemish newspaper) and the BBC were added to the prompt as the topic for the column. In addition, a verbal description of the style of the author was added to the prompt as well.

Figure 3: Example of a generated Amazon product review.

I scooped up this desk cause it looked snazzy, but setting it up was a real chore. The holes in the frame were too tiny for the screws they gave us. I had to wait for my hubby to get back from work to give me a hand, and even he struggled. Once it's all set up, you'll see there's no guard to hold your paper or canvas. They do throw in a guard in the package, but the instructions don't say squat about it. We managed to screw it on, only to find out it's no good. The guard should hold your paper while you're working, but it barely extends beyond the desk. So, if you wanted to use the desk in a raised position (which is why you'd buy a desk that raises, right?), you're gonna have a hard time. Unless you're okay with taping your work to the desk when it's raised, it's not gonna work for you. The desk does look pretty, and the glass is solid. But at this price, I was hoping for better quality and design. These are simple mistakes that the designers could have easily avoided.

Figure 4: Example of a generated Dutch poem.

In de sprankelende zonneschijn, daar ligt de navel in haar pracht, een betoverend klein juweeltje, dat niemand ooit heeft verwacht.

Een herinnering aan het begin, waar het leven ontspringt en groeit, een aanraking van moeder aarde, waar het allemaal voor bloeit.

Oh, wees trots, kleine navel, je bent een wonder van het leven, een symbool van onze connectie, die we nooit zullen opgeven.

Figure 5: Example of a generated Dutch column.

Etienne Vermeersch, de alwetende. Dat Etienne Vermeersch alles weet, weet u. En hij weet het natuurlijk ook, anders wist hij niet alles. Maar wist u dat hij ook alles beter weet? Dat hij, als een soort Vlaamse Socrates, de wijsheid in pacht heeft en ons allen de les leest met een air van superioriteit die zelfs de arrogantste pauw nog zou doen blozen? Ja, Etienne Vermeersch, de man die eruitziet alsof hij net uit een Griekse tragedie is gestapt, compleet met witte baard en alwetende blik, is de onbetwiste koning van de Vlaamse intellectuelen. Hij is de man die altijd gelijk heeft, zelfs als hij ongelijk heeft. Want Etienne Vermeersch weet alles, en wie alles weet, heeft natuurlijk altijd gelijk.

[...]

Dus ja, Etienne Vermeersch weet alles. Maar hij weet vooral heel veel niet. En dat is misschien wel het belangrijkste wat hij zou moeten weten. Want wie denkt dat hij alles weet, weet eigenlijk helemaal niets. En dat is een les die Vermeersch nog moet leren. Maar ja, wie gaat hem dat vertellen? Hij luistert toch niet.

2.2 System Descriptions

The four participating systems include both deep learning models as well as more traditional machine learning approaches leveraging task-specific feature engineering.

2.2.1 DetecTUM[17]

The detection of LLM-generated text is often motivated from the perspective of *application.* For instance, reliable LLM-text detection could serve a real-world purpose, e.g. for authorship verification in an educational context (Dhaini et al. 2023). However, such real-world applications come with ethical issues (Gorichanaz 2023), especially considering the potential of false positives (Dalalah and Dalalah 2023). Rather than focusing on application, our aim for this shared task was to investigate whether we could detect linguistic differences between texts that humans write and texts that machines generate. This line of research has the potential to provide insights into (computational) creativity, for instance by looking into which human writing patterns are rarely replicated by language models.

Motivation We prioritized an explainable approach, in which we could directly estimate the effects of certain linguistic properties. Moreover, given the unique setup of the shared task, namely no ordinary train–dev–test data splits, we set out to create a system[18] that could deal with this data scarcity. One approach to this limitation is to incorporate datasets from other shared tasks or venues. We decided not to choose this direction, since the variation in prompting, models, languages, domains, and formats brings additional challenges. To a lesser extent, these challenges are also presented in the shared task itself: there are two languages, six domains (including a 'mystery' domain), unknown prompting setups, and no information about the model(s) used to generate the samples. These considerations resulted in two main design goals: use *cross-lingual* and *cross-domain* features with the intention to foster transfer learning. This addresses the limited data setup and the decently large number of experimental variables. With these design goals in mind, we extracted linguistic features on two levels: surface level features and parse tree features. Figure 6 presents an overview of the entire system.

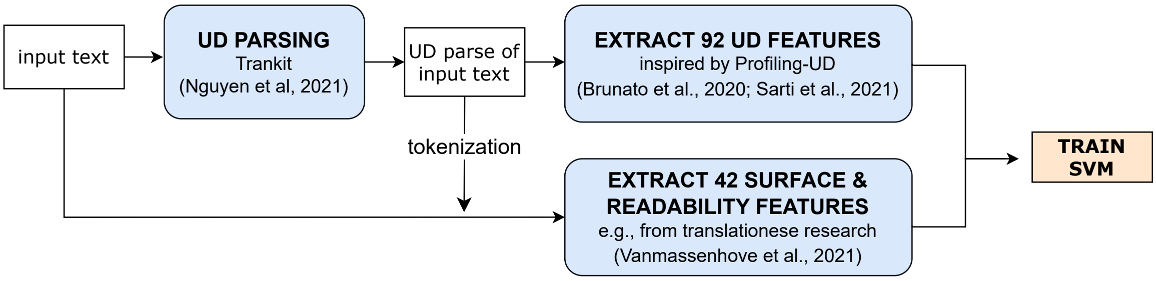
input text
UD PARSING
Trankit
(Nguyen et al, 2021)
UD parse of input text
EXTRACT 92 UD FEATURES
inspired by Profiling-UD
(Brunato et al., 2020; Sarti et al., 2021)
tokenization
EXTRACT 42 SURFACE & READABILITY FEATURES
e.g., from translationese research
(Vanmassenhove et al., 2021)
TRAIN SVM

Figure 6: Pipeline for DetecTUM system.

Surface-level Features The first class of linguistic properties we investigate, are found at the surface-level of a text. That is, they can be extracted more or less directly from the (tokenized) text. Firstly, we examine readability metrics. Readability can be understood as "what makes some texts easier to read than others" (DuBay 2004). The intuition behind this is that humans writing for humans may have a more comprehensive idea of the expectations of their target audience than

[17] Team: Wessel Poelman, Juraj Vladika, Esther Ploeger, Johannes Bjerva & Florian Matthes.

[18] `https://github.com/WPoelman/DetecTUM`

machines generating text for that same audience. These metrics include the Flesch readability score, Gunning Fog, McLaughlin's SMOG and LIX, among others.

Secondly, we draw inspiration from research into another generative NLP task: machine translation. Machine translations have been shown to to exhibit "an exacerbation of frequently observed patterns in combination with a loss of less frequent ones" (Vanmassenhove et al. 2021), also called *machine translationese.* Machine translations have for instance been shown to be less lexically diverse than human translations (Vanmassenhove et al. 2021). Lexical diversity can be measured with type-token ratio or Yule's I for example. We use such features for this shared task, based on the intuition that algorithmic bias may also pertain to other generative large language models.

Parse Tree Features The Universal Dependencies (UD, de Marneffe et al. 2021) framework provides an annotation schema and tools to consistently annotate parts of speech, morphological features, and dependencies relations, across different human languages. At the time of writing, it contains available treebanks for more than 100 languages. UD has been used for numerous NLP applications, including authorship profiling (Brunato et al. 2020, Miaschi et al. 2020). Since UD features are cross-lingual and explicit (i.e., structured output instead of a dense vector representation), they can help uncover writing patterns and other profiling-related characteristics. Sarti et al. (2021) have applied this idea to characterizing *linguistic text complexity* of both humans and language models, for example. They show correlations of syntactic UD features and readability metrics in various settings. Although the shared task is quite different, we can use these features for a similar analysis, namely to discern which syntactic features are most 'human' and which are most 'machine'. Sarti et al. (2021) use *Profiling-UD* (Brunato et al. 2020) to get the UD features. Profiling-UD is software that creates an extended, flat representation of a UD parse tree. For example, the feature `avg_link_len` is the average number of tokens between a head and its dependent, `upos_dist_PUNCT` is the ratio of punctuation tokens over all tokens, `avg_max_depth` is the average maximum depth per parse tree per sentence, and so on. In total there are about 130 features in Profiling-UD, however, this system is not open source.[19] For this reason, we re-implemented most of their system, which resulted in 92 features. The remaining features were left out since they were either ambiguous or lacked information to properly implement. We use Trankit (Nguyen et al. 2021), a recent state-of-the-art UD parser, to get a parse of the input text and extract the features from it.

We train an SVM classifier on the Profling-UD features and the above-mentioned features of the development set of the data. The SVM implementation is from the Python library *sklearn.* After experimenting, we found that the most consistent model was the one trained on all languages and all domains. Language-specific models performed worse, which indicates there is some transfer learning happening. We experimented with an end-to-end DeBERTa-v3 model on just English data, which, surprisingly, resulted in similar results, depending on the genre. It performed better in some configurations, but once we submitted the SVM-based solution to the leaderboard and got first place by a decent margin on English, we decided to stick with our simple approach. Dutch fell behind, but we decided to continue with the intention of characterizing the differences between the texts. Our final submission ended up in second place for both Dutch and English in terms of macro accuracy (Table 5 & 6) and first place for both in terms of macro F1 (Table 9 & 10). Accuracy was used as the deciding measure in the competition.

Results Figure 7 shows the most correlated features with the target label. We filtered these results to only include features that are statistically significant and show both cross-lingual and cross-domain correlations.

Starting with the UD features, we can see that the human-generated texts correlate strongly with the ratio of past tense verb forms (`verbs_tense_dist_Pas, aux_tense_dist_Past`). This could be an artifact of the domains (news, columns, reviews), but it is interesting to see that this apparently *more* pronounced in the human-written texts. The human texts also use more numbers and proper nouns

[19] We contacted the authors, but did not receive any response.

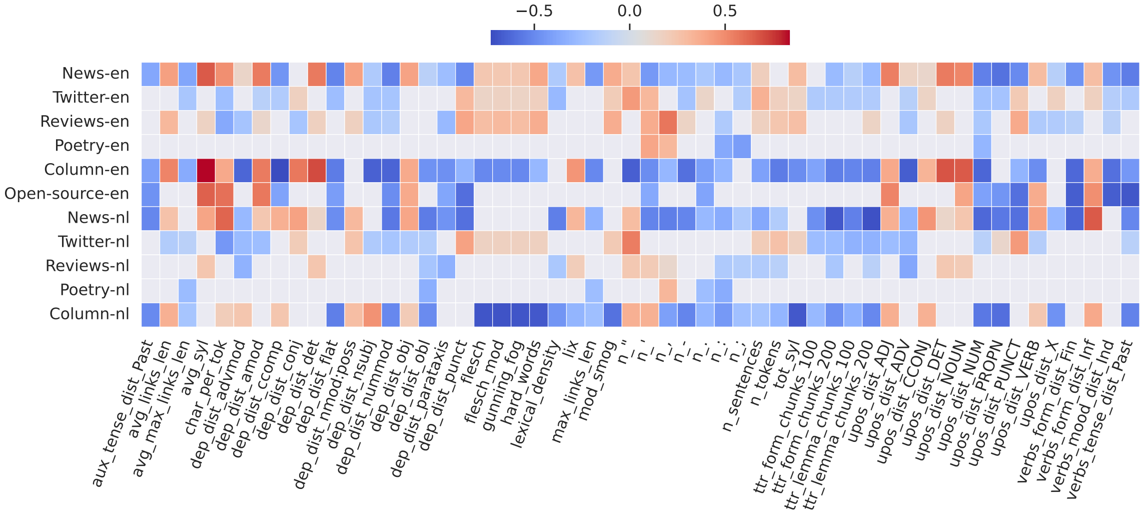

Figure 7: Correlations of features, all values are $p < 0.01$ significant. A positive correlation indicates that an increase in the feature hints towards the text being machine-generated and negative towards human-written texts. An explanation of the features is included in the system repository listed above.

in their writing (`upos_dist_NUM, upos_dist_PROPN`). This might be another artifact of the domains or the language models are more inclined to avoid these given the prompts. When we look at surface level features, it appears that the number of punctuation marks used is quite indicative of human texts, except for quotation marks (`n_*`). This is visible for most domains, but especially for tweets and reviews. This could be an indication that humans are not naturally using a lot of quotation marks in those domains, which makes sense given the length restrictions and informal nature of both domains. Lastly, the readability metrics show interesting patterns (`flesch, flesch_mod, gunning_fog`). On the one hand they strongly negatively correlate with the label for the columns, indicating they hint towards human-written texts. On the other hand, they slightly correlate *positively* with twitter, news, and reviews for English. Readability metrics generally can be interpreted as as *higher is more readable*. This would mean that the column correlation hints towards 'human-written' and 'poorer readability'. The other pattern here hints towards machine-generated news, tweets and reviews being more readable. It is interesting to see that this last effect is *not* present for Dutch. This could be related to the English dominance in the training data of most models.

To conclude, we have shown some interesting linguistic differences in the dataset and our system might prove useful in future characterization efforts. The question whether our approach to the detection task, and even the task itself, is feasible, generalizable, and reliable remains an open question.

2.2.2 Elsevier[20]

The Elsevier team systematically explored various approaches and models in the development of a text classification system. The iterative process began with simple single-model pipelines and evolved into a multi-stage, multi-model system. Through this iterative approach, the team identified the strengths and limitations of each system, informing subsequent decisions.

[20] Team: Yury Kashnitsky & Savvas Chamezopoulos.

Experimental setup In the initial phase, a basic approach was employed using a TF-IDF and Logistic Regression system (Pedregosa et al. 2011a). Trained solely on provided sample data, the system demonstrated strong performance with ”Tweets” but yielded mediocre results with ”News” and ”Reviews” categories. Additionally, a DistilBERT (Sanh et al. 2019) pipeline was tested and validated on the IDMGSP data, producing comparable or superior results to those reported in a relevant paper for Galactica (Taylor et al. 2022) and RoBERTa (Liu et al. 2019). The motivation behind those two approaches was they represented two of the overall best-performing methods in dealing with NLP-related tasks; TF-IDF splits the text on a word level, while transformer-based methods employ token-level segmentation.

To conclude the initial testing phase, an XLMRoBERTa (Conneau et al. 2020) model was trained with the IDMGSP data (Mosca et al. 2023), performing slightly below DistilBERT but outperforming Galactica and RoBERTa (Conneau et al. 2019). For the first system implementation, a single XLMRoBERTa model was selected based on its perceived scalability to multilingual texts across various categories in both English and Dutch. Two versions were trained, one on IDMGSP and another on a larger dataset sourced from 19 different data sources listed in this GitHub repository.

System architecture A single-model approach was insufficient to fully capture the differences among the various text types. Different models' performance was complementary to each other, as the ones that performed best in certain categories were outperformed by others in the rest. Recognizing the limitations of a single-model approach, the team introduced the first iteration of a multi-stage model. This comprised a text type predictor, a single XLMRoBERTa model trained with a dataset containing 5,000 records per text genre described in 2.1.1 (and achieving 96% accuracy at the task of genre prediction), and three text classifiers for specific text types:

- DeepFakeTextDetector (not fine-tuned), a pre-trained classifier described in (Li et al. 2023) – for English News / Poetry / Mystery
- XLMRoBERTa trained with IDMGSP data – for English/Dutch Reviews and Dutch News / Poetry / Mystery
- TF-IDF and Logistic Regression trained directly with CLIN33 dev set – for English / Dutch Tweets

The final submission expanded this multi-stage approach by further splitting the text classifiers:

- DeepFakeTextDetector (not fine-tuned) – for English News / Mystery
- XLMRoBERTa-1 trained with IDMGSP data – for English / Dutch Reviews and Dutch News / Mystery
- XLMRoBERTa-2 trained with 2700 poems by 7 authors and their 2100 chatGPT-generated counterparts (Sawicki et al. 2023) – for English / Dutch Poetry
- TF-IDF and Logistic Regression for English / Dutch Tweets

A full schematic of this system can be found in Figure 8. It is obvious that external data were only used to train the individual components of the system, while during inference, the competition dataset was the only input.

What did not work

- We didn't manage to train anything meaningful with the DeepFake dataset from (Li et al. 2023), probably the context window of 512 tokens is not large enough to capture the differences in human and LLM-generated texts from this particular dataset;
- A somewhat excessively complicated mathematical solution from (Tulchinskii et al. 2023) provided no signal at all; the authors noticed that it does not work in case of high generation temperature, which might be the reason.

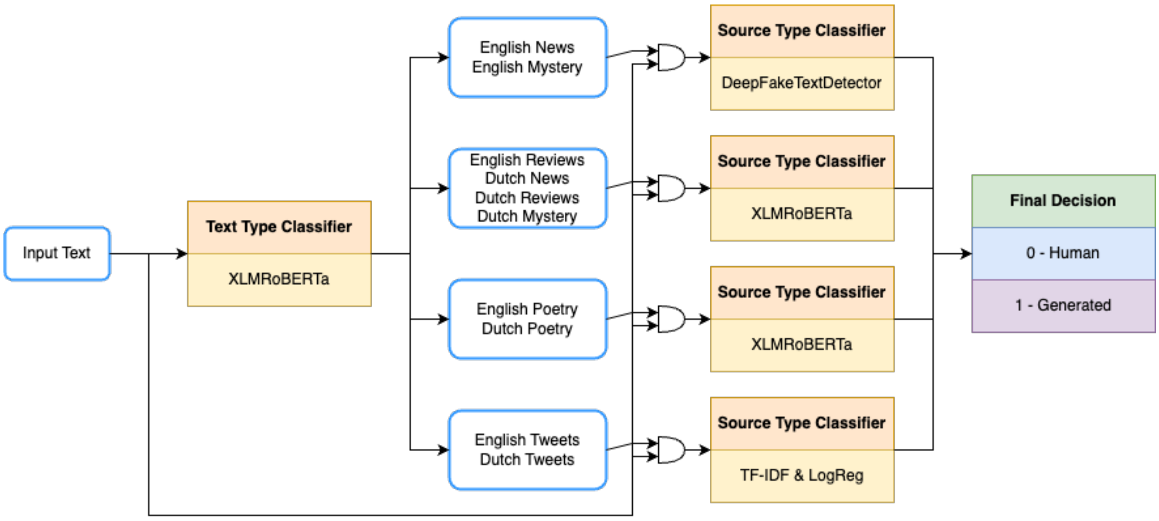

Figure 8: Elsevier system pipeline.

2.2.3 NLP M&S Team[21]

Methodology We have developed a methodology combining various advanced techniques from the text classification domain to address the AI-generated detection task. In addition to the CLIN33 shared dataset, we also utilized another valuable resource for our research: the dataset provided by the AuTexTification[22]: Automated Text Identification shared task; this dataset was introduced to address the issue of binary detection of AI-generated. The dataset contains more than 160000 texts across two languages (English and Spanish) and five domains (tweets, reviews, news, legal, and how-to articles) (Sarvazyan et al. 2023).

Data preprocessing are critical initial phases in machine learning or data analysis tasks. The first step in preprocessing was to convert all textual content to lowercase. By converting all text to lowercase, we can ensure that the same word in different cases is recognized as the same word by the algorithm. Next, we removed elements that contained no information for this task, including URLs, punctuation, and special characters such as mathematical symbols, currency symbols, and other typographical symbols. Removing them helps reduce the data's dimensionality and makes extracting meaningful information easier for the algorithm. Following this, the text was tokenized and lemmatized. Tokenization is the process of dividing the text into individual words or tokens. Lemmatization is a further refinement of this process. It reduces words to their base or root form, grouping different grammatical forms of the same word. This helps reduce the data's complexity and makes it easier for the algorithm to identify patterns.

Data augmentation is a way to add more variety and quantity to training data without collecting new data. This is especially important for this job since we only have a small development dataset and no training data, and it was the most challenging part. This technique is primarily used to prevent overfitting, a common problem where a model performs well on training data but poorly on unseen data. By creating modified versions of the existing data, the model can learn more robust features and generalize better to new data. In this research, we used the following data preparation techniques:

[21] Team: Hadi Mohammadi, Anastasia Giachanou & Ayoub Bagheri.

[22] `https://sites.google.com/view/autextification`

- **Substitution:** This technique involves replacing specific words or phrases in the text with other words or phrases that have a similar meaning. It enhances the model's understanding of context and variations in expression.

- **Deletion:** This technique involves randomly removing words or phrases from the text. It teaches the model to infer meaning or fill in gaps based on context, improving its performance on tasks requiring an understanding of incomplete information.

- **Introducing Spelling Variations:** This method involves deliberately introducing common spelling errors or variations into the text, useful for models used in environments where perfect spelling cannot be guaranteed.

- **Back Translation (English ↔ Dutch):** In this two-step process, a text is first translated from English to Dutch and then back to English, resulting in slightly different wording or structure and thus diversifying the dataset.

- **Paraphrasing with Free Generative AI Models (e.g., GPT-2):** Using AI models like GPT-2 to rephrase text did not significantly increase our model's accuracy, possibly due to their limited rephrasing capabilities. However, more advanced models like GPT-4 might offer more effective paraphrasing.

- **Utilize StratifiedKFold:** This technique involves dividing the dataset into K folds, ensuring each fold is a good representative of the whole. StratifiedKFold maintains the percentage of samples for each class, which is crucial for datasets with imbalanced classes, ensuring that each fold is an accurate representation of the overall class distribution.

- **Address Class Imbalance:** To tackle class imbalance, we employ the following techniques:
  - *RandomOverSampler:* This method involves randomly duplicating examples in the minority class to achieve the balance between the classes.
  - *SMOTE (Synthetic Minority Over-sampling Technique):* This technique generates synthetic samples for the minority class, thereby balancing the dataset in a more nuanced manner than simple duplication.
  - *Compute Class Weights for Balanced Training:* Here, we assign different weights to classes during the training process. More weight is given to minority classes, so the model pays more attention to these classes during training.

Experimental Setup In the training phase, we used the Adam optimizer for training. A learning rate scheduler from callbacks in TensorFlow with a learning rate of 3e-05 and warmup steps of 200 was incorporated to adjust the learning rate during training dynamically. This helps to fine-tune the learning process, often leading to better model performance. An early stopping mechanism was also utilized to prevent needless training once the model's performance ceased to improve significantly. This not only saves computational resources but also helps prevent overfitting. Mixed-precision training was employed to expedite the training process. This method involves using a mix of single-precision (float 32) and half-precision(float16) data types during training, which can significantly reduce the use of computational resources without compromising the model's performance.

Our system leverages a suite of transformers for text processing sourced from the Hugging Face Transformers library. After preprocessing, we set a max length of 256 for tokenization, reflecting the average length of the text data. Hyperparameters were optimized via random search within a defined range of possible values. We examined learning rate values between 1e-5 and 1e-4 and assessed batch sizes of 16, 32, and 64 to determine the optimal balance between computational efficiency and model performance. We also implemented a learning rate scheduler that dynamically adjusts the learning rate according to a cosine decay schedule. The warmup period was set to 200 steps, with incremental steps computed based on the number of epochs and the dataset size. We utilized early stopping based on validation loss to prevent overfitting, with the patience set to 3 epochs. The summary and details of model hyperparameters are shown in Table 4

Table 4: Summary of Model Parameters and Hyperparameters

Parameter	**Description**
Tokenization Max Length	256 tokens
Learning Rate Range	1e-5 to 1e-4 (Default: 3e-5)
Batch Sizes	16, 32, 64
Learning Rate Scheduler	Cosine decay schedule
Warmup Steps	200 steps
Early Stopping Patience	3 epochs
Loss Function	Binary cross-entropy
Optimizer	Adam
Precision Training Policy	Mixed float16

The models were trained using binary cross-entropy loss and the Adam optimizer. We utilized the mixed-precision training policy 'mixed float16' to expedite training without compromising model performance. As demonstrated by (Mohammadi et al. 2023) in their work on ensembling transformer models for detecting online sexism, we utilized a custom function to construct models. This approach allowed us to use the power of different transformers compared to using just one model. Each model utilized a distinct transformer: `bert-base-multilingual-uncased`, `xlm-roberta-base`, and `distilbert-base-multilingual-cased`. Outputs from these transformers were concatenated and passed through a dense layer for binary classification with L2 regularization. In figure 9 the structure of the models that we used is shown.

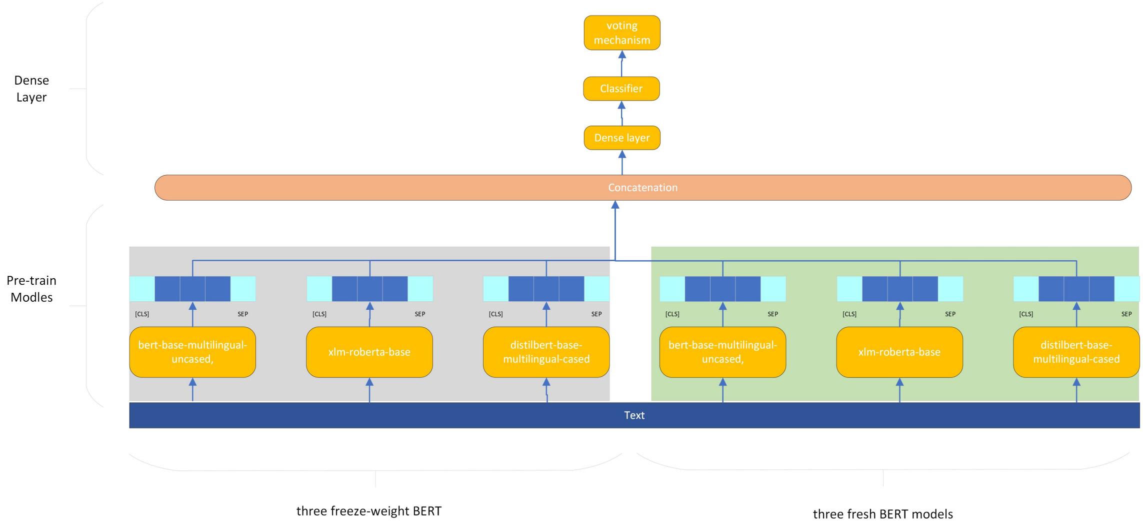

Figure 9: M&S Team model's structure

1. **Training CustomBERT Model Architecture on AuTexTification Train:**

- The CustomBERT model, comprising `bert-base-multilingual-cased`, `xlm-roberta-base`, `distilbert-base-multilingual-cased`, and a dense layer, is trained on the AuTexTification train dataset.

2. **Hyperparameter Optimization and Evaluation:**

    - Hyperparameters of the CustomBERT model are optimized, followed by evaluation on the AuTexTification test dataset. The evaluation focuses on the F1 Score and average metrics across fold divisions.

3. **Freezing Weights of Models Trained on 'AuTexTification Train':**

    - Post-training, the weights of these models are frozen to retain the learned features and prevent overfitting.

4. **Creating a Combined Model:**

    - The combined model includes three freeze-weight BERT models and three fresh BERT models, integrated with a dense layer and a voting mechanism for decision making.

5. **Training the Combined Model on the Augmented Dataset:**

    - This model is then trained on an augmented dataset, created from 90% of the development dataset.

6. **Evaluation on Test Data:**

    - The combined model is evaluated on the untouched 10% of the development dataset.

2.2.4 Van Halteren

Design decisions Our source recognition system was built following a few high level choices. First of all, our existing author recognition system (van Halteren 2022) was not applicable, seeing the inclusion of Twitter, poetry and a mystery genre, which implied short texts of uncertain syntactical nature. This meant building a new system. As development time was short, we added no additional data sets and kept preprocessing minimal. These were necessary but unfortunate choices and will be reconsidered in follow-up research. The core recognition used two central approaches. The first was based on the idea that a human writes texts as a whole, whereas an LLM generates texts word by word. This should lead to a different distribution of information within the text. Here we used standard techniques but with a mostly new set of features, now also to be included in the mentioned existing system. The second was based on simple word counts, as this had shown its value in both literature and the organizers' baseline recognizer. The next choice was based on the idea that every genre is different and should therefore have different distinguishing features. As this implied that a one-model-fits-all strategy should not work, we first applied a genre recognition model and its result determined which source recognition model was applied. Finally, the system was to make ample use of multiple redundancy, in the form of combination of component results, following successes in previous tasks (van Halteren et al. 2001).

Implemented system The minimal preprocessing took the form of lowercasing, splitting on whitespace and removing all but letter and digit characters. The word count model consisted of the frequency vectors for the words in the total development set, split into human and LLM texts, with which the frequency vector of any test text was compared by way of a random forest classifier (combination of five runs; using sci-kit learn (Pedregosa et al. 2011b); sklearn.ensemble RandomForestClassifier).

The features for the more authorship inspired model were mostly related to how information is distributed. Against the background of the text length, there are features for sentence length and length differences between subsequent sentences, both measured in number as well as in total of the IDF of the words, and considering mean, standard deviation and coeffient of variation (CV; standard deviation divided by mean to make it comparable between measurements). IDF lists were based on the written part of the British National Corpus (Burnard and Aston 1998) and SoNaR (Oostdijk et al. 2013). Other features look at the distances between similarly classed words, with classes like low, middle or high IDF words, out-of-vocabulary words, anaphora, and the words themselves. The final word level features look at the frequencies of the top-10 frequency words in the text. At the text level, there are various vocabulary richness measures, with and without correction for text length. And slightly below the text level, the text is split into three equally large parts and there are features for total IDF, differences in IDF and cosine between the parts. Finally, in order to have a model that can be more competitive by itself, features were added for the frequencies of the top-100 most frequent words in the dataset, for frequencies at the character level and one feature for the difference between the cosines of the text with the BOWs for the human and LLM parts of the dev set.

In total, this led to a vector of 374 features. These were used first for genre recognition, again using five runs of the random forest classifier from sci-kit learn, and then for four genre source models, being news, review, Twitter and a generic one using all the development genres together. In the submitted system, a genre specific source model was used when genre recognition confidence was over 0.95. Otherwise, the generic model was used. Application of the source model recognition was in a combination of five runs each of the random forest classifier (sklearn.ensemble RandomForestClassifier), the support vector classification classifier (sklearn.svm SVC) and the multi-layer perceptron classifier (sklearn.neural_network MLPClassifier). All sci-kit learn classifiers were used with default settings. The final result was then combined with the outcome of the word count model. Relative weights for the combination were dependent on the model used and the language in question. Changes between submissions were almost exclusively new weight settings.

Quality measurements Analysis at the macro level showed several interesting things. The genre recognition was far from optimal. News was recognized only about a third of the time for both languages and was therefore mostly handled with the generic model. Reviews were mostly recognized correctly. Twitter was mostly recognized correctly for Dutch, but for English, more than half ended up with generic. Columns were mostly seen as reviews, which had the most impact on quality, but differently for the two languages. Poetry was handled as generic for English, but as Twitter for Dutch. What is unexpected, is that for Dutch this suboptimal genre recognition led to a 0.06 loss on accuracy – spread over all genres – in relation to source recognition using the actual genres, but that for English, it led to a 0.03 gain. For both languages, the impact was mostly caused by the misplaced columns, where the English generic model is better than the review model, but the Dutch review model much better than the generic model. If anything, this demonstrates that generalizing conclusions is a hazardous activity here.

Differences also abound in the relative quality of the models. For English, matters are simplest. For all genres, the trend is that the submission model (SUB) is best, then the word count model (BOW), then the generic model (GEN) and finally the genre-specific model (SPEC). Only for news, SPEC outperformed GEN, and for Twitter, BOW outperformed SUB due to the fact that the Twitter model recognized all test texts as human. For Dutch, there is more variety. News shows BOW best, then SUB, SPEC, GEN. Reviews the same, but with GEN better than SPEC, just like in English. Twitter had BOW best too, but followed by GEN, SUB and SPEC, so here the combination failed. That BOW is not always strong is visible in the new genres columns and poetry. For both, GEN was best, followed by SUB and only then BOW. With hindsight, after seeing these analyses, the best system for Dutch would have been to drop all genre-specific models and combine GEN with BOW, using equal weights. The resulting quality would have been 0.09 better than the submitted system, placing it at first place in the shared task - although other teams would probably also have better scores with hindsight. However, applying this same strategy for English would have lowered quality in relation to the submitted system by 0.035, losing the first place. Here, a good combination would have been SPEC+BOW for news, SPEC+3xBOW for reviews, and GEN+BOW for the rest. These settings lead to a 0.03 improvement over the submitted system. From this analysis, the main conclusion is that a properly working system needs more data, for better training but also for better tuning. Also, genres are different and dedicated systems appear to be needed, although the generic model works remarkably well.

Distinguishing features Analysis at the micro level, as far as conducted so far, already led to a few interesting observations. It must be said that, given the components used, it is almost impossible to determine which features are most useful in the actual classification. The observations here are in terms of the two means and standard deviations of the values of a specific feature, and have mostly been discovered by examining the difference between the means, divided by the sum of the standard deviations, called here the separation measure (SEP). We identify the features with the highest SEP for the various genres, but it should be noted that this is only part of the potential. In the end, it is all features together that do the work. On the other hand, such high-SEP features do show us something about LLM-produced text.

Starting with English News, LLMs generate longer sentences with much less variety in sentence length, more use of lower case characters, lower vocabulary richness and less deviation from a Zipfian word frequency distribution. The CV of the length difference between two adjacent sentences is much higher for LLMs in the test set, but hardly higher in the dev set. For Dutch news, the vocabulary richness and the word frequency distribution has an even stronger presence. Sentence length effects are similar to those for English, including several which are stronger in the test set than in the dev set. In the top only for Dutch is the cosine between the first and the last third of the text, with a mean of around 0.16 for humans and 0.40 for LLMs, and both standard deviations around 0.12. Apparently, humans are adding additional detail towards the end of a news text, while LLMs keep rehashing the same information.

For reviews, the SEP measures are much lower than for news. For both languages, sentence length is gone from the top, but sentence length difference persists. Single quotes appear more frequent in English LLM-generated text, but not in Dutch. Digits also appear in the top, but not consistently between dev and test set, so this is probably just a coincidence in these data sets. All in all, it is surprising that reviews are still recognized at around 0.8, which must be due mostly to the BOW model then. For Twitter, many top features are character related, such as hash tags (more for LLM), quotes (more for LLM) and non-standard characters, most likely emoji (more for humans). Sentence length – whatever this means for Twitter – is also back, following the same pattern as news, but much weaker.

Proceeding to the test-only genres, columns are like news in that they have a number of very strong markers. They also show partly the same trends as news, with humans having higher vocabulary richness, more variety in sentence length and, for Dutch, a lower cosine between beginning and end of the text. New in English are a number of words that are overused by the LLM, with "he", "what", "those", "years", "there" even having a SEP higher than 1. New on the Dutch side is that humans have a higher mean IDF in the middle and end parts of the text. Seeing these strong features suggests that columns should have been recognized with the news model rather than with the review model, which was advised by the genre recognition. For English, this was not needed, as the BOW model saved the day, but for Dutch it would have helped, given a news model score of 0.92, versus BOW 0.58 and reviews 0.67. With poetry, we are back at hardly any strong features. In both languages, humans are using more punctuation, and have larger length differences between adjacent sentences, with less variation. Apart from that, nothing systematic is visible. It is not surprising that poetry was generally hard in the shared task. For the open source system, a similar analysis is not possible, as – obviously – there are no human texts to compare to.

Team name	**Macro-acc.**	News	X	Reviews	Poetry	Columns
Elsevier	**0.75**	0.95	0.70	0.77	0.50	**0.84**
DetecTUM	0.74	0.96	0.74	**0.80**	0.53	0.67
Van Halteren	0.72	**0.97**	0.58	0.78	0.53	0.74
NLP M&S	0.71	0.90	**0.78**	0.70	**0.56**	0.60
Random baseline	0.50	0.50	0.50	0.50	0.50	0.50
Majority voting	0.78	0.98	0.80	0.81	0.53	0.79
Performance ceiling	0.91	1.0	0.95	0.99	0.64	0.95

Table 5: Final results for the Dutch part of the detection task. The reported scores are accuracy scores. Macro-acc. is the average of the accuracy score for each genre. The highest score per category is denoted in bold.

Team name	**Macro-acc.**	News	X	Reviews	Poetry	Columns	Open-source
Van Halteren	**0.85**	**0.99**	**0.69**	**0.82**	0.63	**0.99**	0.96
DetecTUM	0.82	0.93	**0.69**	0.78	**0.80**	0.78	0.92
Elsevier	0.81	0.98	0.65	0.75	0.50	**0.99**	**0.98**
NLP M&S	0.74	0.87	0.63	0.63	0.65	0.85	0.82
Random baseline	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Majority voting	0.87	0.99	0.71	0.85	0.69	0.97	1.0
Performance ceiling	0.95	1.0	0.85	0.99	0.88	1.0	1.0

Table 6: Final results for the English part of the detection task. The reported scores are accuracy scores. Macro-acc. is the average of the accuracy score for each genre. The highest score per category is denoted in bold.

3. Results and Discussion

The 4 participating teams shared the code of their trained models with the organizers. This code was used to run predictions on the test data, which was completely withheld from the participants. The first 2 submissions were also evaluated on a separate public leaderboard, which contained a random subsample of 20% of the test data per genre. Teams were given the choice to choose from these 2 submitted models or to submit a third model, for which they wouldn't know the performance on the leaderboard data.

Tables 5 & 6 show the test accuracies for the participating models. These test scores determined the outcome of the shared task competition (the leaderboard contained the same metrics). We include a random baseline for reference (we don't include a model baseline because of the absence of reference training data). *Majority vote* denotes the performance where we combine all 4 models into an ensemble which generates a positive prediction when at least 2 of the 4 models predict the positive class. *Performance ceiling* is an "oracle" version of this ensemble which considers an instance to be correctly predicted if any of the four models has predicted the correct label. This highlights the proportion of instances which are currently yet out of the reach of any of the participating models.

Since Tables 7 & 8 show that the Phi correlations between the binary predictions of the different models are only moderately strong for both the complete Dutch and English test data, we expected these models to form a good ensemble. However, we see that, while the majority voting model has a modest positive impact on the overall score, there is no substantial impact for any specific genre. This indicates that the high performance ceiling could be caused by random noise which is complementary between the models. The held-out test genre of Dutch poetry is clearly the hardest

	DetecTUM	Elsevier	Hans van Halteren	NLP M&S
DetecTUM	-	0.52	0.54	0.48
Elsevier	0.52	-	**0.65**	0.42
Hans van Halteren	0.54	**0.65**	-	0.45
NLP M&S	0.48	0.42	0.45	-

Table 7: Phi correlations between the different model outputs for the complete Dutch test data. All reported correlations are significant.

	DetecTUM	Elsevier	Hans van Halteren	NLP M&S
DetecTUM	-	0.53	0.61	0.43
Elsevier	0.53	-	**0.73**	0.46
Hans van Halteren	0.61	**0.73**	-	0.53
NLP M&S	0.43	0.46	0.53	-

Table 8: Phi correlations between the different model outputs for the complete English test data. All reported correlations are significant.

for the models, having a ceiling of only 64% accuracy; all other genres where covered very well to almost perfectly.

For comparison, we include Tables 9 & 10 to show the F1 scores instead of the accuracy scores. While this does not greatly impact most of the genres, we see that the performance on English X data as well as Dutch and English poetry declines very strongly, often dropping below the random baseline of 50%. For poetry, it appears the higher accuracy scores were caused by random noise: evaluated using F1, all submissions for Dutch poetry, as well as half of the submissions for English, perform worse than a random baseline. In comparison, the models hold up very well for the equally held-out genre of columns. This is probably caused by its relative similarity to the news genre, for which most models perform almost near-perfectly.

We conclude from all these results that the best-performing models hold up surprisingly well for some of the held-out test data. However, it is also clear that shorter texts (tweets and poetry) pose the most obvious challenge for robust generalization. The main issue here is that we lack a clear indication of what the minimal amount of text should be for LLMs to be detectable. It can already be considered remarkable that LLM-generated tweets are detectable at all, given their 140-character

Team name	**Macro-F1**	News	X	Reviews	Poetry	Columns
Elsevier	0.62	0.95	0.57	0.76	0.00	**0.82**
DetecTUM	**0.63**	0.96	0.72	**0.79**	0.18	0.51
Van Halteren	0.56	**0.97**	0.28	**0.79**	0.11	0.65
NLP M&S	0.62	0.90	**0.79**	0.70	**0.31**	0.39
Random baseline	0.50	0.50	0.50	0.50	0.50	0.50
Majority voting	0.68	0.98	0.77	0.82	0.11	0.73
Performance ceiling	0.86	1.00	0.94	0.99	0.44	0.95

Table 9: F1 scores for the Dutch part of the detection task. Macro-F1 is the average of the F1 score for each genre. The highest score per category is denoted in bold.

Team name	**Macro-F1**	News	X	Reviews	Poetry	Columns	Open-source
Van Halteren	0.79	**0.99**	0.56	**0.83**	0.41	**0.99**	0.96
DetecTUM	**0.81**	0.93	**0.65**	0.80	**0.79**	0.80	0.91
Elsevier	0.70	0.98	0.49	0.78	0.00	**0.99**	**0.98**
NLP M&S	0.67	0.86	0.46	0.52	0.56	0.84	0.80
Random baseline	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Majority voting	0.84	0.99	0.61	0.86	0.59	0.98	1.00
Performance ceiling	0.94	1.00	0.82	0.99	0.86	1.00	1.00

Table 10: F1 scores for the English part of the detection task. Macro-F1 is the average of the F1 score for each genre. The highest score per category is denoted in bold.

limit. While this indicates a strong LLM ”fingerprint” contained within the texts, future work could investigate better generalization of such fingerprints to other short texts.

4. Conclusion

The CLIN33 Shared Task on the Detection of Text Generated by Large Language Models has focused on a cross-domain multilingual binary classification setup which included entirely held-out test genres. As a result, participating teams were given a realistic problem setting for useful detection systems, where potentially generated texts do not come labeled with model, genre, and prompting information. We used both ChatGPT and GPT-4 as well as the open-source Vicuña-13B model to generate a substantial amount of data, which can be used in future research to benchmark detection models.

The participating systems show a diverse range of methodologies and focal points. A common similarity across these systems is their reliance on advanced linguistic and statistical features: DetecTUM and Van Halteren emphasized linguistic patterns and information distribution, while Elsevier and NLP M&S leveraged a combination of traditional and advanced machine learning models like TF-IDF, Logistic Regression, and BERT architectures.

The differences between these systems lie primarily in their experimental setups and specific focus areas. DetecTUM focused on cross-lingual and cross-domain features, aiming to address data scarcity and the varied nature of texts. Elsevier's multi-stage, multi-model approach and NLP M&S's use of data augmentation and transformers highlight a more complex system architecture and preprocessing strategy. Lastly, Van Halteren's system made distinct choices regarding preprocessing and genre-specific modeling, reflecting a more tailored approach to the detection task.

The results of our shared task have shown that this variety of approaches can all be successful for multiple genres, including previously undisclosed test genres. However, these results also show that the main source for future improvement lies in more stable generalization to short texts. While it can be considered remarkable that tweets with a limit of 140 characters can already be classified with above random performance, future research could focus on those edge cases misclassified by all of the participating teams to investigate if a more fundamental ”fingerprint” of LLMs can still be captured. Finally, with the current rise of both closed-source and open-source alternatives to ChatGPT and GPT-4, it will prove crucial to observe whether all these models share any kind of common fingerprint.

References

Antoun, Wissam, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah (2023), Towards a robust detection of language model-generated text: Is ChatGPT that easy to detect?, *in* Servan, Christophe and Anne Vilnat, editors, *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs*, ATALA, Paris, France, pp. 14–27. https://aclanthology.org/2023.jeptalnrecital-long.2.

Brunato, Dominique, Andrea Cimino, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni (2020), Profiling-UD: A Tool for Linguistic Profiling of Texts, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, pp. 7145–7151. https://aclanthology.org/2020.lrec-1.883.

Burnard, Lou and Guy Aston (1998), *The BNC handbook: exploring the British National Corpus*, Edinburgh: Edinburgh University Press.

Chiang, Wei-Lin, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing (2023), Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2019), Unsupervised cross-lingual representation learning at scale, *CoRR*. http://arxiv.org/abs/1911.02116.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2020), Unsupervised cross-lingual representation learning at scale, *in* Jurafsky, Dan, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 8440–8451. https://aclanthology.org/2020.acl-main.747.

Dalalah, Doraid and Osama MA Dalalah (2023), The false positives and false negatives of generative ai detection tools in education and academic research: The case of chatgpt, *The International Journal of Management Education* **21** (2), pp. 100822, Elsevier.

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre, and Daniel Zeman (2021), Universal Dependencies, *Computational Linguistics* **47** (2), pp. 255–308. https://doi.org/10.1162/coli_a_00402.

Dhaini, Mahdi, Wessel Poelman, and Ege Erdogan (2023), Detecting ChatGPT: A survey of the state of detecting ChatGPT-generated text, *in* Hardalov, Momchil, Zara Kancheva, Boris Velichkov, Ivelina Nikolova-Koleva, and Milena Slavcheva, editors, *Proceedings of the 8th Student Research Workshop associated with the International Conference Recent Advances in Natural Language Processing*, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, pp. 1–12. https://aclanthology.org/2023.ranlp-stud.1.

DuBay, William H (2004), The principles of readability., *Online Submission*, ERIC.

Gorichanaz, Tim (2023), Accused: How students respond to allegations of using chatgpt on assessments, *Learning: Research and Practice* **9** (2), pp. 183–196, Taylor & Francis.

Jawahar, Ganesh, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. (2020), Automatic detection of machine generated text: A critical survey, *in* Scott, Donia, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 2296–2309. https://aclanthology.org/2020.coling-main.208.

Kashnitsky, Yury, Drahomira Herrmannova, Anita de Waard, George Tsatsaronis, Catriona Catriona Fennell, and Cyril Labbe (2022), Overview of the DAGPap22 shared task on detecting automatically generated scientific papers, *in* Cohan, Arman, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Drahomira Herrmannova, Petr Knoth, Kyle Lo, Philipp Mayr, Michal Shmueli-Scheuer, Anita de Waard, and Lucy Lu Wang, editors, *Proceedings of the Third Workshop on Scholarly Document Processing*, Association for Computational Linguistics, Gyeongju, Republic of Korea, pp. 210–213. https://aclanthology.org/2022.sdp-1.26.

Keung, Phillip, Yichao Lu, György Szarvas, and Noah A. Smith (2020), The multilingual amazon reviews corpus, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.*

Kosar, Andriy, Guy De Pauw, and Walter Daelemans (2023), Advancing topical text classification: A novel distance-based method with contextual embeddings, *in* Mitkov, Ruslan and Galia Angelova, editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, pp. 586–597. https://aclanthology.org/2023.ranlp-1.64.

Lemmens, Jens, Tess Dejaeghere, Tim Kreutz, Jens Van Nooten, Ilia Markov, and Walter Daelemans (2021), Vaccinpraat: Monitoring vaccine skepticism in dutch twitter and facebook comments, *Computational Linguistics in the Netherlands Journal* **11**, pp. 173–188. https://www.clinjournal.org/clinj/article/view/134.

Li, Yafu, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang (2023), Deepfake text detection in the wild.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019), Roberta: A robustly optimized bert pretraining approach.

Lu, Ning, Shengcai Liu, Rui He, Qi Wang, and Ke Tang (2023), Large language models can be guided to evade ai-generated text detection.

Merkhofer, Elizabeth, Deepesh Chaudhari, Hyrum S. Anderson, Keith Manville, Lily Wong, and João Gante (2023), Machine learning model attribution challenge.

Miaschi, Alessio, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi (2020), Linguistic Profiling of a Neural Language Model, *Proceedings of the 28th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 745–756. https://aclanthology.org/2020.coling-main.65.

Mohammadi, Hadi, Anastasia Giachanou, and Ayoub Bagheri (2023), Towards robust online sexism detection: a multi-model approach with bert, xlm-roberta, and distilbert for exist 2023 tasks, *Working Notes of CLEF.*

Mosca, Edoardo, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh (2023), Distinguishing fact from fiction: A benchmark dataset for identifying machine-generated scientific papers in the LLM era., *in* Ovalle, Anaelia, Kai-Wei Chang, Ninareh Mehrabi, Yada Pruksachatkun, Aram Galystan, Jwala Dhamala, Apurv Verma, Trista

Cao, Anoop Kumar, and Rahul Gupta, editors, *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, Association for Computational Linguistics, Toronto, Canada, pp. 190–207. https://aclanthology.org/2023.trustnlp-1.17.

Nguyen, Minh Van, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen (2021), Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, Online, pp. 80–90. https://aclanthology.org/2021.eacl-demos.10.

Oostdijk, Nelleke, Martin Reynaert, Véronique Hoste, and Ineke Schuurman (2013), The construction of a 500-million-word reference corpus of contemporary written dutch, *Essential speech and language technology for Dutch: Results by the STEVIN programme* pp. 219–247, Springer Berlin Heidelberg.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph (2023), Gpt-4 technical report.

Pedregosa, F., 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 (2011a), Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**, pp. 2825–2830.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. (2011b), Scikit-learn: Machine learning in python, *the Journal of machine Learning research* **12**, pp. 2825–2830, JMLR. org.

Rosati, Domenic (2022), SynSciPass: detecting appropriate uses of scientific text generation, *in* Cohan, Arman, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Drahomira Herrmannova, Petr Knoth, Kyle Lo, Philipp Mayr, Michal Shmueli-Scheuer, Anita de Waard, and Lucy Lu Wang, editors, *Proceedings of the Third Workshop on Scholarly Document Processing*, Association for Computational Linguistics, Gyeongju, Republic of Korea, pp. 214–222. https://aclanthology.org/2022.sdp-1.27.

Sadasivan, Vinu Sankar, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi (2023), Can ai-generated text be reliably detected?

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019), Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, *ArXiv*.

Sarti, Gabriele, Dominique Brunato, and Felice Dell'Orletta (2021), That Looks Hard: Characterizing Linguistic Complexity in Humans and Language Models, *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, Association for Computational Linguistics, Online, pp. 48–60. https://aclanthology.org/2021.cmcl-1.5.

Sarvazyan, Areg Mikael, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso (2023), Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, *arXiv preprint arXiv:2309.11285*.

Sawicki, Piotr, Marek Grzes, Luis Fabricio Góes, Dan Brown, Max Peeperkorn, Aisha Khatun, and Simona Paraskevopoulou (2023), On the Power of Special-purpose GPT Models to Create and Evaluate New Poetry in Old Styles. https://figshare.le.ac.uk/articles/conference_contribution/On_the_Power_of_Special-purpose_GPT_Models_to_Create_and_Evaluate_New_Poetry_in_Old_Styles/24324601.

Taylor, Ross, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic (2022), Galactica: A large language model for science.

Tulchinskii, Eduard, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev (2023), Intrinsic dimension estimation for robust detection of ai-generated texts.

van Halteren, Hans (2022), Automatic authorship investigation, *Language as Evidence: Doing Forensic Linguistics*, Springer, pp. 219–255.

van Halteren, Hans, Jakub Zavrel, and Walter Daelemans (2001), Improving accuracy in word class tagging through the combination of machine learning systems, *Computational linguistics* **27** (2), pp. 199–229, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . .

Vanmassenhove, Eva, Dimitar Shterionov, and Matthew Gwilliam (2021), Machine Translationese: Effects of Algorithmic Bias on Linguistic Complexity in Machine Translation, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, Online, pp. 2203–2213. https://aclanthology.org/2021.eacl-main.188.

Appendix A. Prompt Templates

A.1 News data

A.1.1 Dutch

Your task is to write a news article in Dutch of NUMBER_OF_WORDS words in a human style based on the following headline: ”HEADLINE”.

Make sure to generate a text with low[23] perplexity: you can do this by keeping the vocabulary quite straightforward while using varying grammatical structures and sentence lengths, potentially suboptimal while still correct language.

Return only the generated article, no additional information or title!!!

A.1.2 English

Your task is to write a news article of NUMBER_OF_WORDS words in a human style based on the following headline: ”HEADLINE”.

Make sure to generate a text with low perplexity: you can do this by keeping the vocabulary quite straightforward while using varying grammatical structures and sentence lengths, potentially suboptimal while still correct language.

Return only the generated article, no additional information or title!!!

A.2 X tweets

A.2.1 Dutch

Write a Dutch tweet in a human style using one or more of the following hashtags: LIST_OF_HASHTAGS. Return only the tweet, no additional information or text! Copy the spelling, punctuation and text structure of the following example tweet as closely as possible, even when they are unconventional, but make sure to change the content: ”STYLE_EXAMPLE”

[23] This was an unintended error resulting from cutting and pasting prompt parts, as we actually want the perplexity to be high enough to simulate human-generated text. However, this part of the instruction did not have an actual impact on further lowering the perplexity of the generated texts, while the remaining part of the prompt containing stylistic instructions increased perplexity levels to the point of misleading open-source detectors solely relying on this feature.

A.2.2 English

Write a tweet in a human style using one or more of the following hashtags: LIST_OF_HASHTAGS. Return only the tweet, no additional information or text! Copy the spelling, punctuation and text structure of the following example tweet as closely as possible, even when they are unconventional, but make sure to change the content: "STYLE_EXAMPLE"

A.3 Product Reviews

A.3.1 Dutch

Your task is to write a Dutch book review between 200 to 400 words based on the following review, without copying it: EXAMPLE_REVIEW. You should rewrite this review according to human writing style given in the examples between the <examples> tags.

Please respect all unconventional spelling, orthography and vocabulary. Make sure that you follow the incoherent structure of the reviews as much as possible without improving on the quality of the structure.

Return only the generated review, no additional information or title!!!

Examples of the style: <examples> STYLE_EXAMPLES <examples>

A.3.2 English

Your task is to write an Amazon review between 200 to 400 words based on the following review, without copying it: EXAMPLE_REVIEW. You should rewrite this review according to human writing style given in the examples between the <examples> tags.

Please respect all unconventional spelling, orthography and vocabulary. Make sure that you follow the incoherent structure of the reviews as much as possible without improving on the quality of the structure.

Return only the generated review, no additional information or title!!!

Examples of the style: <examples> STYLE_EXAMPLES <examples>

A.4 Poetry

A.4.1 Dutch

Simple prompts:

1. Je bent een begenadigd dichter. Schrijf een kort gedicht.
2. Je bent een begenadigd dichter. Schrijf een haiku.
3. Je bent een begenadigd dichter. Schrijf een limerick.

Elaborate prompts:

1. Schrijf een kort, contemplatief gedicht. Gebruik geen voornaamwoorden. Kies een concreet onderwerp, zoals een fietsbel, of een ruitenwisser.
2. Je bent een bekend Nederlands dichter. Schrijf een kort gedicht over een verlaten station in het avondlicht.
3. Je bent een bekend Nederlands dichter. Schrijf een elegie over het vervlieden van het plattelandsleven.
4. Je bent een bekend Nederlands dichter. Schrijf een kort gedicht over ochtendmist in de wijngaard.
5. Schrijf een limerick in het Nederlands. Gebruik Tilburg als plaatsnaam.

6. Schrijf een kort gedicht over een vervallen speeltuin.
7. Schrijf een gedicht over hoe je je zou voelen als zwaartekracht plotseling zou verdwijnen en dan onverwacht weer terugkeert.
8. Schrijf een gedicht over de vredige ervaring van het luisteren naar regen op een tentdak tijdens het kamperen.
9. Schrijf een gedicht over hoe internetmemes troost bieden in tijden van eenzaamheid en verdriet.
10. Schrijf een ode aan een koffiezetapparaat.
11. Schrijf een ode aan fietsen in de stad.
12. Schrijf een elegie over het afscheid van een bitterbal.
13. Schrijf een limerick over een bejaarde rolschaatser.
14. Schrijf een lofdicht over de erwt.
15. Schrijf een gedicht over de letterzetter, die schade toebrengt aan een boom.
16. Schrijf een limerick over een man die geen dromen of verbeelding heeft.
17. Schrijf een gedicht over de impact van overtoerisme op historische steden.
18. Schrijf een korte ode (2 coupletten) aan Oostende.

A.4.2 English

Simple prompts:

1. You are a renowned poet. Write me a short poem.
2. You are a renowned poet. Write me a haiku.
3. You are a renowned poet. Write me a limerick.
4. You are a renowned poet. Write me a vilanelle.

Elaborate prompts:

1. Write me a haiku on the poetry of a plastic bag in the wind.
2. Write me a haiku about the feeling you get when you just missed the last train.
3. You are a renowned poet. Write me a sonnet on people that jump the queue at a bakery.
4. Write me a short poem. Pick an interesting and uncommon theme, not a cliché one like nature or love. No depths of the ocean either. Use a topic like croissants or honeymoon arguments.
5. Write me a sonnet about a dog and a pelican discussing yesterday's night out.
6. Write me a haiku on chameleons.
7. Write me a short rhyme poem on violins and happy endings.
8. Write me a short, contemplative poem. Pick a random topic, like napkins or toothpicks. Do not use any pronouns.
9. Write a haiku about an unanswered phone call.
10. Write a poem in an ABAB ABAB scheme about any topic you like, but preferably something far-fetched that has to do with student life.
11. Write a poem in an ABAB ABAB scheme about really strange hobbies.
12. Write me a short poem. Pick an interesting and uncommon theme, not a cliché one like nature or love. No depths of the ocean either. Don't make it happy go lucky. It needs to be mysterious.

13. Write an ode to the typewriter.
14. Write a haiku about the effects of social media.
15. Write a sonnet about the art of glassblowing.
16. Write a limerick about a toad. Use Oxford as a placename.
17. Write a sonnet about Harry Potter.
18. Write a haiku about rust.
19. Write a limerick about driving in the rain. End the first line with Munich.
20. Write a sonnet about the sensation of your foot falling asleep.
21. Write a humorous limerick about an overweight dachshund.
22. Write a villanelle about trying a new food for the first time.
23. Write an elegy from the perspective of a raindrop.
24. Write a humorous poem about a professor who suffers from flatulence.

A.5 Columns

A.5.1 Dutch

You are KaAIman, a Flemish columnist. Your task is to write a column in Dutch of at least 500 words. Your style is humorous, biting, mocking, insulting, vicious, vitriolic, and satirical. Examples of your style are in the texts between the <examples> tags. The text should be based on the topic between the <topic> tags.

Examples of the style: <examples>EXAMPLES<examples>

Topic: <topic>TOPIC<topic>

A.5.2 English

You are Laura Kuenssberg, a UK columnist. Your task is to write a column.

Important: the text should be more than 1000 words long.

Your style is clear and concise, analytical, balanced, engaging, immediate, in-depth, and conversational.

Examples of your style are in the texts between the <examples> tags.

The text should be based on the topic between the <topic> tags.

The first sentence should be a title.

Examples of the style: <examples>EXAMPLES<examples>

Topic: <topic>TOPIC<topic>