International Journal of Speech Technology
https://doi.org/10.1007/s10772-024-10143-3

Classification of human- and AI-generated texts for different languages and domains

Kristina Schaaff[1] · **Tim Schlippe**[1] · **Lorenz Mindner**[1]

Received: 10 August 2024 / Accepted: 1 September 2024
© The Author(s) 2024

Abstract
Chatbots based on large language models (LLMs) like ChatGPT are available to the wide public. These tools can for instance be used by students to generate essays or whole theses from scratch or by rephrasing an existing text. But how does for instance a teacher know whether a text is written by a student or an AI? In this paper, we investigate *perplexity*, *semantic*, *list lookup*, *document*, *error-based*, *readability*, *AI feedback* and *text vector* features to classify *human-generated* and *AI-generated* texts from the educational domain as well as news articles. We analyze two scenarios: (1) The detection of text generated by AI from scratch, and (2) the detection of text rephrased by AI. Since we assumed that classification is more difficult when the AI has been prompted to create or rephrase the text in a way that a human would not recognize that it was generated or rephrased by an AI, we also investigate this *advanced prompting* scenario. To train, fine-tune and test the classifiers, we created the *Multilingual Human-AI-Generated Text Corpus* which contains *human-generated*, *AI-generated* and *AI-rephrased* texts from the educational domain in English, French, German, and Spanish and English texts from the news domain. We demonstrate that the same features can be used for the detection of *AI-generated* and *AI-rephrased* texts from the educational domain in all languages and the detection of *AI-generated* and *AI-rephrased* news texts. Our best systems significantly outperform GPTZero and ZeroGPT—state-of-the-art systems for the detection of *AI-generated* text. Our best *text rephrasing* detection system even outperforms GPTZero by 181.3% relative in F1-score.

Keywords Generative AI · ChatGPT · Natural language processing · Features · Prompting · Artificial intelligence · Text classification

1 Introduction

In recent years, chatbots based on large language models (LLMs) have rapidly gained prominence, integrating seamlessly into various aspects of daily life (Pelau et al., 2021). Engineered to mimic human-like interactions, these digital assistants offer support, provide information, and even daily companionship (Dibitonto et al., 2018; Arteaga et al., 2019; Falala-Séchet et al., 2019; Adiwardana et al., 2020). Amongst the multitude of chatbots, OpenAI's ChatGPT[1] has distinguished itself as a leading tool for generating text, attracting over one million users within merely five days of its launch (Taecharungroj, 2023).

✉ Kristina Schaaff
kristina.schaaff@iu.org

✉ Tim Schlippe
tim.schlippe@iu.org

[1] IU International University of Applied Sciences, Bad Honnef, Germany

As chatbots such as ChatGPT become increasingly integrated into our daily routines, it becomes even more crucial to distinguish between *human-generated* and *AI-generated* text[2]. While both types of text can transport information, a fundamental difference lies in the underlying intent: *Human-generated* text is usually created with the explicit purpose of transporting a message, while *AI-generated* text is produced by algorithms engineered to emulate human-like writing based on statistical probabilities. *AI-generated* text may exhibit repetitive expressions, while *human-generated* text tends to be more original and imaginative. Furthermore, texts generated by chatbots often lack reliability which is often also referred to as hallucinations (Ji et al., 2023). As chatbots advance, recognizing *AI-generated* text accurately becomes increasingly challenging.

In the educational domain, chatbots pose serious problems, including plagiarism by having them rephrase existing

[1] https://chat.openai.com

[2] We will focus on LLM-based chatbots only as they are more powerful than traditional chatbots.

Published online: 04 December 2024

text or generate texts from scratch such as essays. Consequently, there is a strong need for tools that can differentiate between these *AI-generated* and *AI-rephrased* texts in educational scenarios. As chatbots can quickly produce large volumes of text on various topics, it is also easy to create numerous fake news and propaganda articles that seem genuine (Thompson, 2023). This poses a risk of readers falling into filter bubbles. Hence, the development of systems capable of distinguishing between *human-* and *AI-generated* texts is also of great importance when it comes to the detection of *AI-generated* news articles.

Our goal was to collect a text corpus that can be used to train, fine-tune, and test systems that recognize (1) text that was generated entirely by an AI (*AI-generated*), and (2) text that was rephrased by an AI based on an existing text (*AI-rephrased*). The presented results build upon the work presented in Mindner et al. (2023) and Schaaff et al. (2023), where a variety of novel features such as *text subjectivity*, *list lookup features*, and *error-based features* were investigated for identifying English (*EN*), Spanish (*ES*), German (*DE*), and French (*FR*) text from the educational domain produced by ChatGPT. These languages were chosen due to their widespread global use (Ethnologue, 2023), highlighting the need for a multilingual approach in the ongoing effort to detect *AI-generated* text.

To evaluate how well our features perform in another domain, we moreover analyze the detection of *AI-generated* news articles using our features. While other research concentrates on fake news detection through methods like fact-checking, our objective is to identify *AI-generated* news that can be applied as a feature in fake news detection systems.

Consequently, our contributions are as follows:

- We investigate two scenarios: (1) The detection of text generated by AI from scratch (*AI-generated*), and (2) the detection of text rephrased by AI (*AI-rephrased*).
- We demonstrate, that the same features can be used for the detection of *AI-generated* and *AI-rephrased* text in the languages *EN*, *FR*, *DE*, and *ES*.
- We demonstrate, that the same features developed for texts from the educational domain can successfully be applied to the detection of *AI-generated* news articles.
- We collected the *Human-AI-Generated Text Corpus* which includes texts from two domains as well as the four languages *EN*, *FR*, *DE* and *ES*.
- To contribute to the improvement of the detection of *AI-generated* text, we share our corpus with the research community.[3]
- Our best systems significantly outperform GPTZero and ZeroGPT—state-of-the-art systems for the detection of *AI-generated* text.

We decided to use ChatGPT for our research as at the time of our research it was the most widely used chatbot to generate texts and is publicly available.

In Sect. 2, we will give an overview of related work concerning ChatGPT and the detection of *human-generated* and *AI-generated* text. In Sect. 3, we will provide more details about the text corpus we created to train, fine-tune and test our systems. We will present our analyzed features in Sect. 4. In Sect. 5, we will describe the experimental setup. The experiments and results of our study will be presented in Sect. 6. We will conclude our work in Sect. 7 and give an outlook on future research.

2 Related work

In this section, we will describe the related work concerning ChatGPT as well as the classification of *human-* and *AI-generated* texts.

2.1 ChatGPT

Since its release by OpenAI in late 2022, ChatGPT has revolutionized the field of AI (Mesko, 2023), and several other chatbots such as Google's Gemini[4] or Meta's Llama[5] (Touvron et al., 2023) have been released. Those tools are capable of generating text in response to user queries across a wide range of domains. Its successful implementation is demonstrated in areas like education (Baidoo-Anu & Owusu Ansah, 2023), medicine (Jeblick et al., 2023), and language translation (Jiao et al., 2023). ChatGPT is based on the Generative Pre-trained Transformers (GPT) model and fine-tuned through reinforcement learning with human feedback. This approach empowers ChatGPT to comprehend the significance and purpose behind user prompts, enabling it to provide relevant and helpful answers. Although the exact quantity of training data remains undisclosed, it is worth noting that the predecessor of GPT-3—a model significantly larger than other language models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and T5 (Roberts, 2019)—was trained with 175 billion parameters and 499 billion crawled text tokens (Brown et al., 2020). Through extensive training on a diverse dataset, ChatGPT has developed a nuanced comprehension of human language, enabling it to produce text closely mirroring human writing (Mitrović

[3] https://github.com/iu-ai-research/human-AI-generatedTextCorpus

[4] https://gemini.google.com/

[5] https://ai.meta.com/llama

et al., 2023). It is even able to express empathy in several aspects (Schaaff et al., 2023).

For these reasons, in this work, we decided to evaluate our features for the detection of text *AI-generated* and *AI-rephrased* by ChatGPT.

2.2 Detecting human-generated and AI-generated texts

Various tools, such as GPTZero (Shrivastava, 2023), ZeroGPT [6], AI Content Detector[7], and GPT-2 Output Detector[8] (Mitchell et al., 2023), have been developed to recognize *AI-generated* text. Additionally, ongoing research focuses on establishing new datasets for this purpose and determining which features and classifiers enhance classification accuracy. For instance, (Yu et al., 2023) present a dataset comprising both *human-generated* and *AI-generated* abstracts, primarily exploring commercial and non-commercial systems. However, in their study, they solely focus on *EN*. Recent investigations include approaches to detect *AI-generated* text, including XGBoost (Shijaku & Canhasi, 2023), decision trees (Zaitsu & Jin, 2023), and transformer-based models (Mitrović et al., 2023; Guo et al., 2023). Mitrović et al. (2023) assessed attributes of *AI-generated* text sourced from *EN* customer reviews, constructing a transformer-based classifier achieving 79% accuracy. Zaitsu and Jin (2023) accomplished 100% accuracy in Japanese text detection using decision trees, integrating stylometric features such as bigrams, comma position, and function word rates. Guo et al. (2023) analyzed features of *human-* and *AI-generated* responses in *EN* and Chinese, employing fine-tuned RoBERTa models (Liu et al., 2019) to achieve 98.8% and 96.4% F1-scores for *EN* and Chinese responses, respectively. Shijaku and Canhasi (2023) addressed essay detection in *EN*, proposing an XGBoost model achieving 98% accuracy using TF-IDF-generated features and manual feature sets. Soni and Wade (2023) analyzed *human-* and *AI-generated* text summaries, achieving a 90% accuracy via DistilBERT[9] (Sanh et al., 2019). Mindner et al. (2023) explored features to identify *AI-generated* and *AI-rephrased* text in *EN*. They achieved F1-scores of 96% and 78% for *AI-generated* and *AI-rephrased* text, respectively, across diverse topics—even when they used *advanced prompting* that instructs the chatbot to produce text that sounds as if it was *human-generated*.

To the best of our knowledge, our study is the first to investigate an extensive range of features and classifiers across multiple languages and domains. Given the current absence of multilingual *human-generated* and *AI-generated* text datasets, we expanded our *Human-AI-Generated Text Corpus* (Mindner et al., 2023) to encompass *EN*, *FR*, *DE*, and *ES*. Our study includes XGBoost (Shijaku & Canhasi, 2023), Random Forrest (RF) (Breiman, 2001), and Multilayer Peceptrons (MLPs) (Murtagh, 1991). We compare our results against two state-of-the-art AI text detection tools: GPTZero and ZeroGPT. GPTZero, employed by over one million users (Shrivastava, 2023), shows reliable outcomes primarily for *EN* texts. Consequently, we used ZeroGPT as a reference for *FR*, *DE*, and *ES*.

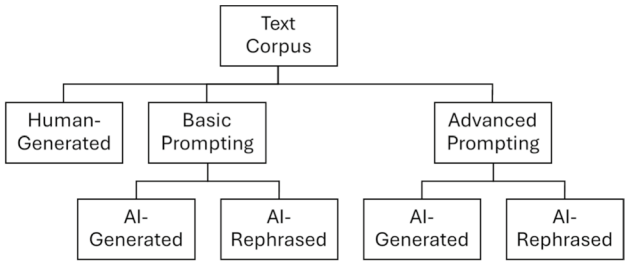

Fig. 1 Structure of our text corpus

3 Our multilingual human-AI-generated text corpus

Developing a system for the classification of *human-generated* and *AI-generated* texts in several languages requires a text corpus that is suitable for our task. Therefore, we developed the *Multilingual Human-AI-Generated Text Corpus* which includes both *human-generated* and *AI-generated* texts in *EN*, *FR*, *DE*, and *ES*. The following sections will describe the methodology we used to construct our text corpus.

3.1 Structure of the corpus

Our primary goal was to generate a text corpus that can be used to train, fine-tune, and test systems that differentiate between *human-generated* and *AI-generated* text. Figure 1 illustrates the structure of our text corpus. As the way a prompt is written has a major influence on the generated text, we did not only analyze the detection of text generated with *basic prompting* but also with *advanced prompting*, i.e. text generated using additional instructions indicating the text has to be the way a human would write it. Moreover, chatbots cannot only be used to *generate* text from scratch but also to *rephrase* existing texts. The ability to identify *AI-rephrased* text is also an important issue, especially if it comes to plagiarism. Therefore—besides *AI-generated* texts—our text corpus also includes texts which

[6] https://www.zerogpt.com

[7] https://copyleaks.com/ai-content-detector

[8] https://openai-openai-detector--mqlck.hf.space

[9] https://huggingface.co/docs/transformers/model_doc/distilbert

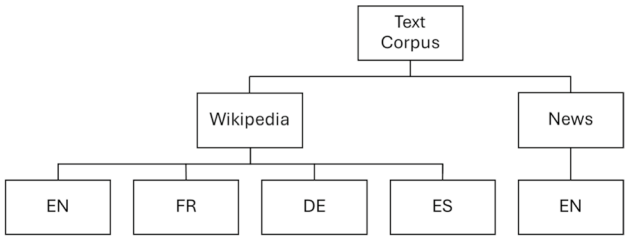

Fig. 2 Domains and languages of our text corpus

were *AI-rephrased* based on *human-generated* texts. The process of selecting *human-generated* text is described in Sect. 3.2. Producing text using *basic prompting* is illustrated in Sect. 3.3. Section 3.4 describes how we generated texts using *advanced prompting*.

Our text corpus includes texts from two different domains and four languages. Figure 2 demonstrates the domains and languages of our text corpus. The first domain is articles taken from Wikipedia. This domain was chosen as the way the texts are written is similar to essays or homework in the educational domain which is an important application area of *AI-generated* text detection. The process of how we selected the articles is described in Sect. 3.2.1. Additionally, our goal was to analyze texts from different categories to ensure that the performance of the features and systems developed are generalizable amongst different types of texts which is reflected in the different categories, such as *history* and *sports*, we selected for both domains. The second domain covers news articles as the detection of generated news can be an important contribution to fake news detection. Section 3.2.2 provides more details about how we selected respective news articles. In addition to analyzing the performance to detect *AI-generated* texts on our two domains, another goal was to evaluate the performance among languages. Therefore, we selected Wikipedia articles that were available in the following languages: *EN*, *FR*, *DE*, and *ES*.

3.2 Human-generated texts

We collected *human-generated* text (1) for the purpose of teaching *AI-generated* and *AI-rephrased* detection systems the characteristics of *human-generated* text, and (2) as a basis to retrieve *AI-rephrased* text. As introduced in Sect. 3.1, our text corpus includes *human-generated* texts from two domains: Wikipedia and news texts. From both domains, we selected texts from different categories to ensure that our developed features generalize amongst different kinds of texts.

3.2.1 Wikipedia texts

When developing our text corpus, one important goal was to provide texts that are written similar to essays or homework in education. Therefore, we decided to build our text corpus based on articles from Wikipedia. Accordingly, we created a corpus that spans a wide array of categories relevant to such settings. We defined the following text categories: *biology*, *chemistry*, *geography*, *history*, *IT*, *music*, *politics*, *religion*, *sports*, and *visual arts*, selecting ten topics within each category. The selected articles had to be available in *EN*, *FR*, *DE*, and *ES*. To make sure that the Wikipedia articles were not *AI-generated*, we chose only articles that had been written before November 2022. As text from the educational domain tends to be longer and consists not only of a few sentences, the minimum length for *human-generated* texts was defined to be 1000 characters including white spaces.

Table 1 shows the number of words, paragraphs, and sentences for the *human-generated* Wikipedia texts for every language and the respective text category. We observe that the paragraphs contain different numbers of sentences and words in different languages: For instance, while *EN*, and *FR* have the same number of paragraphs (*P*) overall, *FR* texts have approximately 30% fewer sentences (*S*) and approximately 20% fewer words (*W*).

Table 1 *Human-generated* Wikipedia article statistics (P = #paragraphs, S = #sentences, W = #words)

	EN			FR			DE			ES		
Category	P	S	W	P	S	W	P	S	W	P	S	W
Biology	44	188	3739	42	109	2670	32	101	1736	39	124	3065
Chemistry	44	167	3590	35	105	2413	34	127	2291	49	137	3446
Geography	35	167	3386	45	135	3037	33	90	1655	42	125	3381
History	43	189	4578	47	183	4810	33	129	2417	45	129	4023
IT	40	141	2916	37	89	2290	36	106	1700	48	134	3525
Music	39	191	4177	38	111	2936	29	122	2133	35	137	3964
Politics	43	172	4298	43	137	3363	40	146	2455	47	132	4272
Religion	40	171	3796	42	110	2779	34	115	2049	40	123	3379
Sports	51	204	4692	51	167	4330	41	154	2550	65	190	5517
Visual arts	36	147	3165	35	92	2352	23	90	1486	40	129	3378
Overall	**415**	**1737**	**38337**	**415**	**1238**	**30980**	**335**	**1180**	**20472**	**450**	**1360**	**37950**

Table 2 Distribution of news articles by source and category

Source	Category					Overall
	Crime	Entertainment	Politics	Science	Sports	
Axios	2	0	2	3	0	7
CNN	4	2	8	7	3	24
Economist	1	0	0	2	1	4
Mashable	1	0	1	0	1	3
People	5	11	0	2	0	18
Reuters	1	1	5	2	6	15
The New York times	4	4	1	1	7	17
The Verge	2	2	0	1	1	8
Vice	0	0	0	1	1	2
Washington post	0	0	0	1	1	3
Overall	**20**	**20**	**20**	**20**	**20**	**100**

Table 3 *Human-generated* news article statistics (P = #paragraphs, S = #sentences, W = #words)

Category	P	S	W
Crime	20	442	8520
Entertainment	20	426	7244
Politics	20	412	9095
Science	20	503	10276
Sports	20	606	11000
Overall	**100**	**2389**	**46135**

3.2.2 News texts

For the *human-generated* news texts, we collected 100 news articles from the *All The News 2.0* dataset[10] provided by the research project *Components* (2023), equally distributed across the following five categories: *Crime*, *entertainment*, *politics*, *science*, and *sports*. We selected texts from different sources ensuring some variability in the way they are written. For every category, we selected 20 articles. The number of articles from each news source over our six news categories is listed in Table 2.

Table 3 summarizes the number of words, paragraphs, and sentences for the *human-generated* news articles across our news categories. Each category consists of 20 news articles, each consisting of one paragraph. When we compare the number of sentences with the *human-generated EN* Wikipedia texts summarized in Table 1, we observe that the news articles consist of 38% more sentences and 20% more words, indicating that in our text corpus the news articles are on average longer than the Wikipedia articles.

[10] https://components.one/datasets/all-the-news-2-news-articles-dataset

3.3 Basic AI-generated and AI-rephrased texts

Our goal was to collect a text corpus that can be used to train systems that recognize (1) text that was generated entirely by an AI (*AI-generated*), and (2) text that was rephrased by an AI based on an existing text (*AI-rephrased*). In the following paragraphs, we will present our *basic prompting* to retrieve *AI-generated* and *AI-rephrased* texts for the Wikipedia domain and the news domain.

3.3.1 Wikipedia texts

To generate 100 *EN AI-generated* texts across our 10 Wikipedia categories, we instructed ChatGPT with the following *basic prompt*: “Generate a text based on the topic: <topic>.” For obtaining 100 *EN AI-rephrased* texts for each category, we used the following *basic prompt*: “Rephrase the provided text: <text extracted from Wikipedia article>”. An example for both *EN* prompts is shown in Fig. 3. Table 4 gives an overview of the number of paragraphs, sentences and words of our *EN human-generated* texts compared to the *EN basic AI-generated* and *AI-rephrased* texts in Wikipedia style. Additionally, we used the prompts demonstrated in Table 5 to generate the *basic AI-generated* and *AI-rephrased* Wikipedia texts for *FR*, *DE* and *ES*.

3.3.2 News texts

The *basic AI-generated* news articles were produced using the prompt “Generate a news article on the following topic: <news title>”. We did not provide any additional instructions. This way, we generated a total of 100 *basic AI-generated* news articles. For the *AI-rephrased* news articles, we used the following prompt: “Rephrase the following news article: <text from news article>” Table 6 summarizes the number of paragraphs, sentences, and words for the *basic*

Fig. 3 Example prompt and ChatGPT's response for *basic EN* text *generation* & *rephrasing*

Text Generation:

Prompt: "Generate a text on the following topic: Vienna"

ChatGPT: "Vienna, the capital of Austria, is a city rich in history, culture, and entertainment. Located in the heart of Europe, it has been a melting pot of various cultures [...]"

Text Rephrasing:

Prompt: "Rephrase the following text: Vienna is the capital, largest city, and one of nine states of Austria. Vienna is Austria's most populous city, with about two million inhabitants, and its cultural, economic, and political center. [...]"

ChatGPT: "Vienna, the capital and largest city of Austria, is a center for culture, economy, and politics, with a population of almost two million people. [...]"

Table 4 *Basic EN AI-generated* & *AI-rephrased* Wikipedia articles (P = #paragraphs, S = #sentences, W = #words)

	Human			AI-generated			AI-rephrased		
Category	P	S	W	P	S	W	P	S	W
Biology	44	188	3739	54	139	2500	21	96	1899
Chemistry	44	167	3590	56	140	2684	28	129	2539
Geography	35	167	3386	60	167	3006	27	114	2540
History	43	189	4578	61	148	3017	26	146	3205
IT	40	141	2916	51	129	2624	24	91	1872
Music	39	191	4177	53	154	2701	27	137	2900
Politics	43	172	4298	56	131	2866	25	104	2341
Religion	40	171	3796	51	138	2684	25	108	2409
Sports	51	204	4692	59	143	2904	30	128	2913
Visual arts	36	147	3165	54	136	2686	22	85	2024

Table 5 Prompts for *basic* text *generation* & *rephrasing*

Language	Prompt
Text generation	
EN	Generate a text on the following topic: <topic>
FR	Rédigez un texte sur le thème suivant: <topic>
DE	Erstelle einen Text zum folgenden Thema: <topic>
ES	Genera un texto sobre el siguiente tema: <topic>
Text rephrasing	
EN	Rephrase the following text: <topic>
FR	Reformulez le texte suivant: <topic>
DE	Formuliere den folgenden Text um: <topic>
ES	Reformule el siguiente texto: <topic>

AI-generated and *AI-rephrased* news articles across our news categories.

3.4 Advanced AI-generated and rephrased texts

Another objective was to explore the scenario where the AI was prompted to generate or rephrase the text in an *advanced* way that was not distinguishable from *human-generated* text. In the following sections, we will present our *advanced prompting* to generate *AI-generated* and *AI-rephrased* texts for the Wikipedia domain and the news domain.

3.4.1 Wikipedia texts

To obtain 100 *advanced AI-generated* texts, we prompted ChatGPT with "Generate a text on the following topic in a way a human would do it: <topic>" for each of the 10 topics within each category. Subsequently, we created 100 *advanced AI-rephrased* texts by instructing ChatGPT to "Rephrase the following text in a way a human would do it: <text from Wikipedia article>" for each of the 10 topics within each category. Figure 4 illustrates how we instructed ChatGPT to generate a text about Vienna and how we instructed ChatGPT to rephrase the Wikipedia text about Vienna. Table 7 gives an overview of the number of paragraphs, sentences, and words of our *EN human-generated* Wikipedia texts compared to the *EN advanced AI-generated* and *AI-rephrased* Wikipedia texts. We used the prompts shown in Table 8 to generate the *advanced AI-generated* Wikipedia texts for *FR*, *DE* and *ES*.

Table 6 *Basic AI-generated* & *AI-rephrased* news articles (P = #paragraphs, S = #sentences, W = #words)

	AI-generated			AI-rephrased		
Category	P	S	W	P	S	W
Crime	163	367	6901	21	145	3113
Entertainment	151	359	6667	20	124	2677
Politics	153	341	6680	20	134	2956
Science	149	297	6440	20	139	3157
Sports	159	347	7044	20	127	2708
Overall	**775**	**1711**	**33752**	**101**	**669**	**14611**

Fig. 4 Example prompt and chatGPT's response for *EN advanced* text *generation* & *rephrasing*

Text Generation:

Prompt: "Generate a text on the following topic in a way a human would do it: Vienna"

ChatGPT: "Vienna is a beautiful and historic city located in the heart of Austria. It is the capital city and largest metropolitan area in the country. Known for its classical music, fine arts, and stunning architecture, Vienna is a [...]"

Text Rephrasing:

Prompt: "Rephrase the following text in a way a human would do it: Vienna is the capital, largest city, and one of nine states of Austria. Vienna is Austria's most populous city, with about two million inhabitants, and its cultural, [...]"

ChatGPT: "Vienna, located in Austria, is the country's capital, largest city and one of its nine states. With a population of about two million, it is Austria's most populous city and serves as its cultural, [...]"

3.4.2 News texts

The *advanced AI-generated* news articles were produced using the prompt "Generate a news article on the following topic in a way a human would do it: <news title>". In total, we generated 100 *advanced AI-generated* news articles. For the *rephrased AI-generated* news articles, we used the following prompt: "Rephrase the following news article in a way a human would do it: <text from news article>". Table 9 summarizes the number of paragraphs, sentences, and words for the *advanced AI-generated* and *AI-rephrased* news articles across our news categories.

4 Our features for the classification of human-generated and AI-generated texts

In this section, we will present the 8 feature categories which we analyzed for the classification of *human-generated* and *AI-generated* texts, consisting of 37 features that

Table 7 *Advanced EN AI-generated* & *AI-rephrased* Wikipedia articles (P = #paragraphs, S = #sentences, W = #words)

	Human			AI-generated			AI-rephrased		
Category	P	S	W	P	S	W	P	S	W
Biology	44	188	3739	47	111	2057	18	79	1487
Chemistry	44	167	3590	48	124	2374	22	103	1859
Geography	35	167	3386	49	136	2575	24	106	2028
History	43	189	4578	52	132	2583	22	113	2415
IT	40	141	2916	51	128	2538	16	86	1541
Music	39	191	4177	48	128	2426	21	109	2145
Politics	43	172	4298	52	127	2672	25	111	2300
Religion	40	171	3796	48	122	2623	28	120	2488
Sports	51	204	4692	53	143	2685	31	118	2407
Visual arts	36	147	3165	43	119	2238	23	95	1931

Table 8 Prompts for *advanced* text *generation* & *rephrasing*

Language	Prompt
Text generation	
EN	Generate a text on the following topic in a way a human would do it: <topic>
FR	Rédigez un texte sur le thème suivant comme le ferait un un être humain: <topic>
DE	Erstelle einen Text zum folgenden Thema, so wie ein Mensch es tun würde: <topic>
ES	Genera un texto sobre el siguiente team de la format en queue lo haria un humano: <topic>
Text rephrasing	
EN	Rephrase the following text in a way a human would do it: <Wikipedia text>
FR	Refurmuléz le texte suivant comme le ferait un un être humain: <Wikipedia text>
DE	Formuliere den folgenden Text so um, wie es ein Mensch tun würde: <Wikipedia text>
ES	Reformula el siguiente texto de la format en queue lo haria un humano: <Wikipedia text>

were originally introduced in Mindner et al. (2023). While many of these features have been used in previous studies, in Mindner et al. (2023) we also introduced 10 new features. A summary of all features is shown in Table 10.

4.1 Perplexity features

Perplexity estimates a language model's predictive ability for word sequences (Vu et al., 2010), with lower values indicating better performance in predicting the next word. *AI-generated* texts, relying on statistical patterns, often show lower perplexity due to lower word variability compared to *human-generated* texts. When differentiating *AI-generated* and *human-generated* texts, a significant distinction lies in the unpredictability and variety of *human-generated* text. Through their creativity, knowledge, and experiences, humans can create texts with novel word pairings, concepts, and frameworks. *AI-generated* texts are typically derived from statistical models and regulations, making them more foreseeable and prone to repetition. Our *perplexity-based* features are based on Mindner et al. (2023); Gehrmann et al. (2019); Mitrović et al. (2023); Guo et al. (2023) and calculated using the *evaluate package*[11] and the English GPT-2 model.[12]

In our research, we focused on two perplexity metrics: First, we used the average perplexity (PPL_{mean}), which is determined by averaging the perplexities of all sentences within a given text corpus. Additionally, we examined the highest perplexity value (PPL_{max}), which reflects the maximum perplexity encountered. In both cases, the perplexities were calculated based on the probabilities given by a language model. For that, we used the language model GPT-2[13], consistent with Guo et al. (2023) and Mitrović et al. (2023).

Figure 5 illustrates the distribution of PPL_{mean} across *human-generated*, *AI-generated*, and *AI-rephrased* texts for *EN* on the *Wikipedia* texts. A significant proportion of *AI-generated* texts displays perplexity values around 25, indicating lower perplexities compared to *human-generated* texts, which often show perplexities near 50. *AI-rephrased* texts do not follow this trend, suggesting that the classification using the PPL_{mean} feature may encounter more difficulties with *AI-rephrased* texts than with purely *AI-generated* ones.

4.2 Semantic features

Semantic features refer to the properties of words or phrases used to represent their meanings. Mitrović et al. (2023) and Guo et al. (2023) successfully used these features for the classification of *human-generated* and *AI-generated* texts. For the implementation, we used the sentiment analysis of TextBlob.[14] To the best of our knowledge, in Mindner et al. (2023) we were the first to analyze the degree of objectivity and subjectivity as a feature to classify *human-generated* and *AI-generated* texts.

Sentiment analysis involves the automated identification of sentiment within text, categorizing it into classes like *negative*, *neutral*, or *positive* (Wankhade et al., 2022), or assigning a *sentiment score*. The application of sentiment analysis on text aids in determining whether it is *human-generated* or *AI-generated*, as highlighted in Mitrović et al. (2023) and Guo et al. (2023).

Our analysis includes two semantic characteristics: Initially, we utilized the sentiment analysis feature of TextBlob,[15] a Python library designed for processing textual data, to obtain a sentiment polarity score ($sentiment_{polarity}$) ranging

[11] https://github.com/huggingface/evaluate

[12] https://huggingface.co/gpt2

[13] https://github.com/openai/gpt-2

[14] https://textblob.readthedocs.io/en/dev/quickstart.html

[15] https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis

Table 9 *Advanced AI-generated & AI-rephrased* news articles (P = #paragraphs, S = #sentences, W = #words)

	Generated			Rephrased		
Category	P	S	W	P	S	W
Crime	132	281	5419	20	150	2847
Entertainment	133	280	5415	20	177	2410
Politics	133	277	6071	20	106	2590
Science	125	263	5523	20	112	2548
Sports	116	263	5253	20	123	2651
Overall	**639**	**1364**	**27681**	**100**	**608**	**13046**

from -1 to +1. Here, a score of -1 indicates an extremely negative text, whereas +1 signifies an extremely positive text. Secondly, we employed the same library to acquire a subjectivity score ($sentiment_{subjectivity}$) spanning from 0 to +1, with 0 denoting a highly objective text and +1 indicating a highly subjective text.

Figure 6 illustrates based on *EN Wikipedia* texts that *AI-generated* and *AI-rephrased* texts tend to have higher $sentiment_{subjectivity}$ scores in comparison to *human-generated* texts. This observed distribution might be attributed to the fact that ChatGPT was further optimized through reinforcement learning based on human feedback (Natalie, 2023).

4.3 List lookup features

Using our *ListLookup* features, we examined word or character features, such as stop words or special characters. These features have been employed by Shijaku and Canhasi (2023); Kumarage et al. (2023). We generated a list of discourse markers and personal pronouns using ChatGPT, which was subsequently reviewed by language experts. Stop word counts were calculated using the Natural Language Toolkit (NLTK) (Bird & Loper, 2004). For example, if a word is found in a stop word list (e.g., "a", "an", "the", "of"), we know that it is a stop word.

Consequently, we used both features for our classification. As a new promising list lookup feature, we included the number of discourse markers ($discourseMarker_{count}$), such as "however", "furthermore", or "moreover". Additionally, we took the absolute and relative numbers of repetitions of the article's title ($titleRepetition_{count}$, $titleRepetition_{relative}$) since we detected that *AI-generated* text often repeats keywords from the title.

Figure 7 visualizes the feature $specialChar_{count}$ based on the *EN Wikipedia* texts as a representative of the *list lookup features*. The figure indicates that the $specialChar_{count}$ is more widely distributed when the text is *human-generated*.

4.4 Document features

Our *document* features encompass content and structural elements within a document, including word frequencies, syntactic structures, and corpus statistics. These features have been successfully used by Kumarage et al. (2023); Shijaku and Canhasi (2023); Guo et al. (2023); Mitrović et al. (2023); Zaitsu and Jin (2023). Since in Kumarage et al. (2023) the standard deviation of words and sentences performed well, we also included the standard deviation of the number of unique words per sentence ($uniqWordsPerSentence_{stdev}$) as a new feature. In addition, the number of quotation marks ($quotation_{count}$) is used, as we found that AI produces fewer quotation marks. For *sentence-* and *word-related* features, we initially segmented the text into sentences and words using NLTK's `sent_tokenize` and `word_tokenize` functions. Regarding part-of-speech, we used the NLTK function `pos_tag`.

For instance, Fig. 8 highlights for the *Wikipedia* texts in *EN* that *AI-generated* and *AI-rephrased* texts are characterized by a minimal presence of quotation marks, with more than 80% and 60% of such texts respectively containing no quotation marks at all. *Human-generated* texts exhibit a greater frequency of quotation marks, with over 20% of the examples containing one quotation mark and more than 15% including four quotation marks.

4.5 Error-based features

Our analysis revealed that *AI-generated* texts typically contain fewer spelling and grammar mistakes compared to *human-generated* texts. Consequently, in Mindner et al. (2023) we introduced a novel category of features which we name *error-based features*, within which we evaluated the number of spelling and grammar mistakes ($grammarError_{count}$) as well as the appearance of multiple consecutive blank spaces ($multiBlank_{count}$) as features from this category. For the identification of spelling and grammar errors, we employed *LanguageTool*[16], an open-source tool known for its integration with OpenOffice as a spellchecker. The detection of multiple blanks was accomplished using regular expressions.

[16] https://github.com/jxmorris12/language_tool_python

Table 10 Summary of our features for the classification of generated texts

Category	Feature	Description	Reference
Perplexity	PPL_{mean}	mean PPL	(Gehrmann et al., 2019; Mitrović et al., 2023; Guo et al., 2023)
	PPL_{max}	maximum PPL	(Gehrmann et al., 2019; Mitrović et al., 2023; Guo et al., 2023)
Semantic	$sentiment_{polarity}$	degree of positivity/negativity [-1,+1]	(Mitrović et al., 2023; Guo et al., 2023)
	$sentiment_{subjectivity}$	degree of subjectivity [0,+1]	new
List Lookup	$stopWord_{count}$	number of stop words	(Shijaku & Canhasi, 2023)
	$specialChar_{count}$	number of special characters	(Kumarage et al., 2023)
	$discourseMarker_{count}$	number of discourse markers	new
	$titleRepetition_{count}$	absolute repetitions of title	new
	$titleRepetition_{relative}$	relative repetitions of title	new
Document	$wordsPerParagraph_{mean}$	ø number of words per paragraph	(Kumarage et al., 2023)
	$wordsPerParagraph_{stdev}$	stdev of *wordsPerParagraph*	(Kumarage et al., 2023)
	$sentencesPerParagraph_{mean}$	ø number of sentences per paragraph	(Kumarage et al., 2023)
	$sentencesPerParagraph_{stdev}$	stdev of *sentencesPerParagraph*	(Kumarage et al., 2023)
	$wordsPerSentence_{mean}$	ø number of words per sentence	(Kumarage et al., 2023)
	$wordsPerSentence_{stdev}$	stdev of *wordsPerSentence*	(Kumarage et al., 2023)
	$uniqWordsPerSentence_{mean}$	ø number of unique words per sentence	(Shijaku & Canhasi, 2023)
	$uniqWordsPerSentence_{stdev}$	stdev of *uniqWordsPerSentence*	new
	$words_{count}$	number of running words	(Guo et al., 2023; Shijaku & Canhasi, 2023; Kumarage et al., 2023)
	$uniqWords_{count}$	number of unique words	(Kumarage et al., 2023)
	$uniqWords_{relative}$	relative number of unique words	(Kumarage et al., 2023)
	$paragraph_{count}$	number of paragraphs	(Kumarage et al., 2023)
	$sentence_{count}$	number of sentences	(Kumarage et al., 2023)
	$punctuation_{count}$	number of punctuation marks	(Kumarage et al., 2023)
	$quotation_{count}$	number of quotation marks	new
	$character_{count}$	number of characters	(Kumarage et al., 2023)
	$uppercaseWords_{relative}$	relative number of words in uppercase	(Shijaku & Canhasi, 2023)
	$personalPronoun_{count}$	absolute number of personal pronouns	(Mitrović et al., 2023)
	$personalPronoun_{relative}$	relative number of personal pronouns	(Mitrović et al., 2023)
	$POSPerSentence_{mean}$	ø number of unique POS-tags/sentence	(Guo et al., 2023; Kumarage et al., 2023; Zaitsu & Jin, 2023)
Error-Based	$grammarError_{count}$	number of spelling/grammar errors	new
	$multiBlank_{count}$	number of multiple blanks	new
Readability	*fleschReadingEase*	Flesch Reading Ease score [0-100]	(Shijaku & Canhasi, 2023; Flesch, 1948)
	fleschKincaidGradeLevel	Readability as U.S. grade level [0-100]	(Shijaku & Canhasi, 2023; Kincaid et al., 1975)
AI Feedback	*AIFeedback*	Ask AI if text was generated by AI	new
Text Vector	*TF-IDF*	500-dim TF-IDF vector of 1-/2-grams	(Shijaku & Canhasi, 2023; Solaiman et al., 2019)
	Sentence-BERT	ø Sentence-BERT vector	(Reimers & Gurevych, 2019)
	Sentence-BERT-dist	ø distance of Sentence-BERT vectors	new

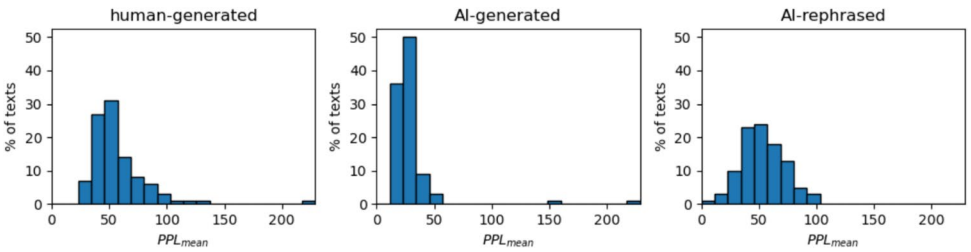

Fig. 5 PPL_{mean} distribution for *EN*

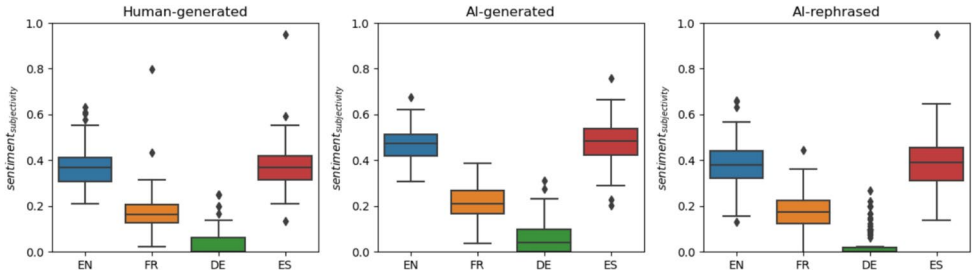

Fig. 6 *sentiment*$_{subjectivity}$ distribution for *EN*

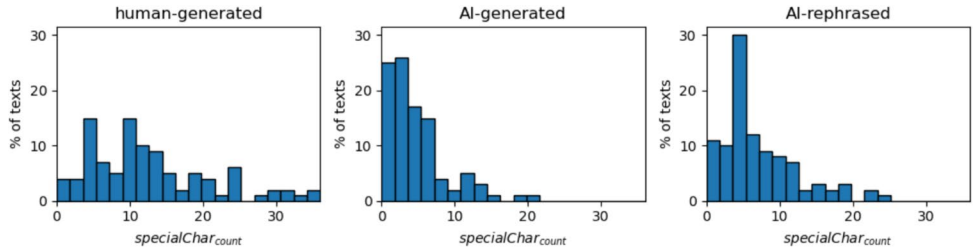

Fig. 7 *specialChar*$_{count}$ distribution for *EN*

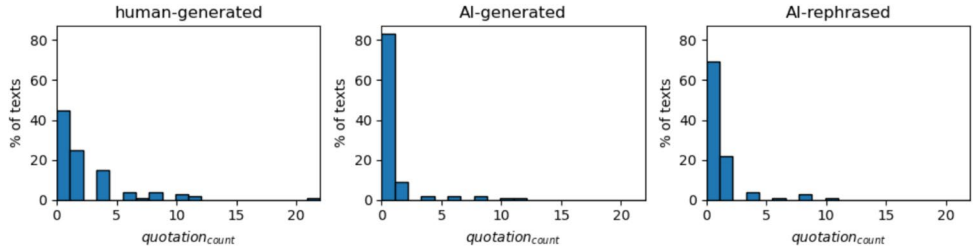

Fig. 8 *quotation*$_{count}$ distribution for *EN*

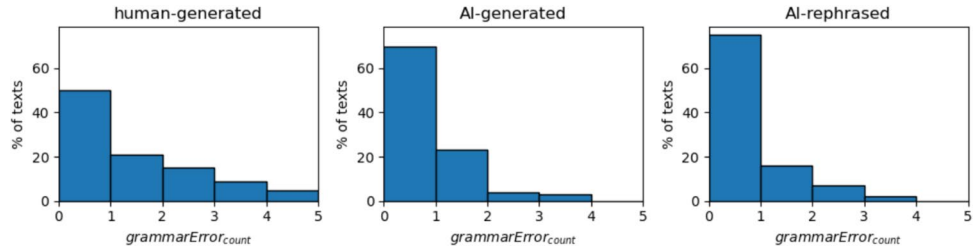

Fig. 9 *grammarError*$_{count}$ distribution for *EN*

Figure 9 shows the distribution of the feature *grammarError*$_{count}$ across *EN human-generated*, *AI-generated*, and *AI-rephrased Wikipedia* texts. From this, it is evident that *LanguageTool* identifies a higher number of spelling and grammar errors in *human-generated* texts compared to both *AI-generated* and *AI-rephrased* texts.

4.6 Readability features

Readability features evaluate the readability level of texts as done in Flesch (1948); Kincaid et al. (1975); Shijaku and Canhasi (2023) using Flesch Reading Ease and Flesch-Kincaid Grade Level. We use the Python library *Textstat*[17] to compute these metrics. *Textstat* provides functions for evaluating text statistics, grade level, complexity, and overall readability.

In alignment with Shijaku and Canhasi (2023), we use the *Flesch Reading Ease* score (*fleschReadingEase*) and the *Flesch-Kincaid Grade Level* (*fleschKincaidGradeLevel*) in our study. The *Flesch Reading Ease* is an index of text readability, where higher scores denote easier readability and lower scores suggest more complex texts (Flesch, 1948). The *Flesch-Kincaid Grade Level* offers a grade-level equivalence for the text's comprehensibility, facilitating its assessment by educators, guardians, and librarians for suitability (Kincaid et al., 1975). Both scores are derived from a formula that takes into account the total number of words, sentences, and syllables in the text (Flesch, 1948; Kincaid et al., 1975).

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

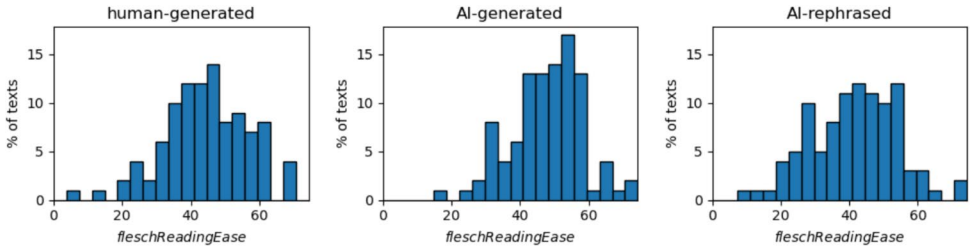

Fig. 10 *fleschKincaidGrade-Level* distribution for *EN*

Table 11 Prompts used for *AI-Feedback* generation

Language	Prompt
EN	Was the following text generated by ChatGPT?
FR	Le texte suivant a-t-il été généré par ChatGPT?
DE	Wurde der folgende Text von ChatGPT generiert?
ES	¿El siguiente texto fue generado por ChatGPT?

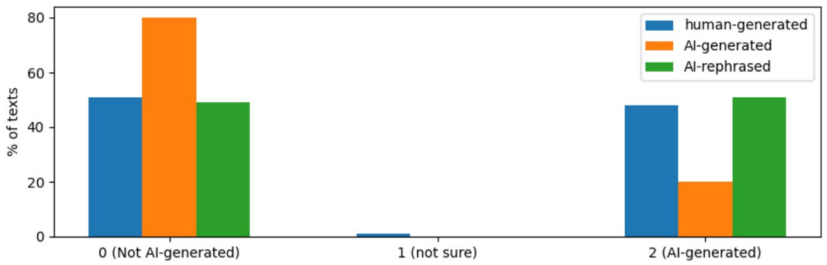

Fig. 11 *AIFeedback* distribution for *EN*

Figure 10 shows the *Flesch-Kincaid Grade Level* score distribution for *EN* in our *Wikipedia* data set. While a higher number of *AI-generated* texts is on a higher level between 50–60, the *fleschKincaidGradeLevel* distributions of *human-generated* and *AI-rephrased* text look comparable.

4.7 AI feedback features

Another novel feature introduced in Mindner et al. (2023) is the *AI feedback feature*. It reflects how an AI categorizes the text (Mindner et al., 2023).

To obtain this feature, we asked ChatGPT with a prompt in the respective language, if it generated the text. The prompts we used for this purpose are shown in Table 11. If ChatGPT answers 'yes', the value 2 is assigned to the feature. If it states that it did not generate the text, we assign the value of 0. If ChatGPT answers that it is not sure, the value 1 is assigned.

Figure 11 shows the distribution of the *AIFeedback* feature for the *Wikipedia* text corpus in *EN*. It shows that the feature does not seem to discriminate between *AI-generated* and *AI-rephrased* text.

4.8 Text vector features

Our *TextVector* features analyze the semantic content of a text. First, we converted the texts into *TF-IDF* vectors which performed well in Shijaku and Canhasi (2023) and Solaiman et al. (2019). Furthermore, to take advantage of the benefits of the semantic vector space, we also converted the texts into *Sentence-BERT* word embeddings (Reimers & Gurevych, 2019). Finally, as we observed that *AI-generated* and *AI-rephrased* texts often contain similar sentences due to repetitive phrases or patterns, we converted each sentence of an article into a Sentence-BERT vector and computed the average distance of all vectors (*Sentence-BERT-dist*). The lower the distance for this third *TextVector* feature, the higher the chance that a text is *AI-generated* or *AI-rephrased* since similar sentences are closer in the semantic vector space. For this purpose, we used the Sentence-BERT implementation `distiluse-base-multilingual-cased-v2`[18]. This implementation does not only support our four languages but also more than 50 other languages, guaranteeing reliable results for possible future research.

[18] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

5 Experimental setup

In this section, we will present our experimental setup.

5.1 Selected classifiers

We selected XGBoost (Shijaku & Canhasi, 2023), Random Forest (Breiman, 2001), and Multilayer Perceptron (Murtagh, 1991) for our experiments which we will describe in the following sections.

5.1.1 XGBoost

XGBoost is a commonly used classification algorithm and was also trained by Soni and Wade (2023), Kumarage et al. (2023), as well as by Shijaku and Canhasi (2023) to classify *human-generated* and *AI-generated* text. Therefore, we also used XGBoost for our research. The XGBoost classifier was implemented using the `xgboost` Python library[19]. To guarantee reproducible results, we set the classifierâ€™s random state to 42. By defining the same random state for all trainings, we initialized XGBoost always identically. Thus, only the data used to train the classifier and the hyperparameters have an impact on the results.

5.1.2 Random forest

The Random Forest (RF) classifier was used in the research of Zaitsu and Jin (2023) to classify whether a Japanese expert text was *AI-generated* or *human-generated*. Furthermore, Kumarage et al. (2023) employed RF. As they achieved good results using RF, we also made use of this algorithm. For the implementation, we leveraged the `RandomForestClassifier` provided by `scikit-learn`[20]. As with XGBoost, we set the random state for RF to 42, which controls both how the features are sampled in the search for the best split at each node, and the randomness in the creation of the trees. Thus, repeatable results can be guaranteed for the same set of hyperparameters and training data.

5.1.3 Multilayer perceptron

While none of the research focusing on *human-generated* and *AI-generated* text classification used Multilayer Perceptrons (MLPs) for classification, we intended to test this deep learning-based approach to evaluate its usefulness compared to the traditional approaches. For the implementation, we used the deep learning API Keras[21], which provides a simple API to enable fast model development and has a strong adoption among data scientists, according to a study conducted by Kaggle (Mooney, 2022). An MLP consists of an input layer, multiple hidden layers, and an output layer representing the classification result. The input layer takes the features derived from the texts. The number of hidden layers depends on the hyperparameters.

5.1.4 Training and testing of the classifiers

To provide stable results, we performed a 5-fold cross-validation, randomly dividing our corpus in each fold into 80% for training, 10% as a validation set to optimize the hyperparameters, and an unseen test set containing 10% of the texts. Each set contained 50% *human-generated* and 50% *AI-generated* or *AI-rephrased* texts.

Hyperparameter tuning was performed for each classifier to minimize the risk of obtaining poor results due to inappropriate hyperparameters. XGBoostâ€™s hyperparameters influence the regularization, complexity, and training of the classifier (Chen & Guestrin, 2016). The hyperparameters for the RF influence the complexity, e.g., regarding the number of trees and the maximum depth of the trees (Breiman, 2001). The hyperparameters for the MLP influence the depth and complexity of the neural network. A grid search was applied, with every possible combination of hyperparameters being trained. Due to the varying number of features used to train the classifiers and the related difference in complexity of the optimization problem, the aim for the selection of the hyperparameters was to provide a wide range of possible hyperparameter values.

5.2 Tools for the detection of AI-generated text as references

We selected two tools designed for detecting *AI-generated* texts as references: GPTZero and ZeroGPT. These tools were state-of-the-art at the time of our research. GPTZero even indicates a user base exceeding 1 million (Shrivastava, 2023). However, we observed that GPTZero's outcomes were only reliable for *EN* texts. Therefore, we utilized ZeroGPT as the reference for *FR*, *DE*, and *ES*.

6 Experiments and results

In this section, we will describe our results across the 8 feature categories, presented in Sect. 6.1.1, using the three classifiers XGBoost, RF, and MLP, which we described in

[19] https://github.com/dmlc/xgboost

[20] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[21] https://keras.io

Table 12 Results for *basic EN AI-generated* Wikipedia articles: XGBoost vs. RF vs. MLP ($Acc_{GPTZero}$=76.0%, $F1_{GPTZero}$=78.9%)

	XGBoost		RF		MLP	
Category	Acc	F1	Acc	F1	Acc	F1
Perplexity	83.0%	82.2%	87.0%	85.3%	82.0%	82.1%
Semantic	72.0%	72.9%	75.0%	75.6%	73.0%	72.3%
ListLookup	83.0%	82.8%	80.0%	81.1%	81.0%	82.9%
Document	90.0%	90.9%	93.0%	93.3%	97.0%	97.0%
ErrorBased	55.0%	61.7%	55.0%	61.7%	56.0%	63.9%
Readability	60.0%	56.3%	63.0%	59.3%	60.0%	56.8%
AIFeedback	62.0%	67.1%	62.0%	67.1%	62.0%	68.1%
TextVector	90.0%	89.9%	95.0%	94.9%	81.0%	80.6%
All	90.0%	90.9%	**98.0%**	**98.0%**	87.0%	87.8%

Sect. 5.1. As in other studies (Guo et al., 2023; Kumarage et al., 2023; Mitrović et al., 2023), we assessed the quality of our classification using accuracy (*Acc*) and F1-score (*F1*). In Sect. 6.1, we will compare how well our features classify text created with the *basic* and *advanced prompting* for Wikipedia texts from the educational domain. Then, we will analyze the multilingual performance of our features in *EN*, *FR*, *DE*, and *ES* in Sect. 6.2. Finally, we will evaluate how well our features perform in the news domain in Sect. 6.3.

6.1 Classification of human- and AI-generated texts generated with basic and advanced pompting

In this section, we will compare how well our features classify our Wikipedia texts, created with the *basic* and *advanced prompting*.

6.1.1 Basic prompting

Tables 12 and 13 present the *Acc* and *F1* for detecting *EN AI-generated* and *AI-rephrased* Wikipedia texts which were created using *basic prompting*, without additional instructions. The best results are marked in bold.

As illustrated in Table 12, the best-performing feature categories are the *Document* features and the *TextVector* features. With 97% *Acc* and 97% *F1*, *Document* performs substantially better with MLP than XGBoost and RF, while *TextVector* is most successful with RF (*Acc*=95.0%, *F1*=94.9%). Most of our systems are able to outperform GPTZero ($Acc_{GPTZero}$=76.0%, $F1_{GPTZero}$=78.9%). Our best system with *All* features even performs better than GPTZero by 28.9% relative in *Acc* and 24.2% relative in *F1*.

Table 13 shows that the performance of the *basic text rephrasing* detection system is consistently worse than the performance of the *basic text generation* detection system. The feature categories which perform best are *TextVector* (*Acc*=79.0%, *F1*=78.2%), *ListLookup* (*Acc*=72.0%, *F1*=73.7%), *Document* (*Acc*=72.0%, *F1*=70.9%). In general, XGBoost performs better than RF and MLP. All of our systems outperform GPTZero ($Acc_{GPTZero}$=43.0%, $F1_{GPTZero}$=27.8%). Compared to GPTZero, our best system *TextVector* (*Acc*=79.0%, *F1*=78.2%) performs much better by 83.7% relative in *Acc* and even 181.3% relative when looking at *F1*.

6.1.2 Advanced pompting

In addition to the *basic prompting*, we analyzed the effect of *advanced prompting* designed to *generate* or *rephrase* text in a way a human would do it on the classification performance.

Table 13 Results for *basic EN AI-rephrased* Wikipedia texts: XGBoost vs. RF vs. MLP ($Acc_{GPTZero}$=43.0%, $F1_{GPTZero}$=27.8%)

	XGBoost		RF		MLP	
Category	Acc	F1	Acc	F1	Acc	F1
Perplexity	52.0%	48.7%	55.0%	54.6%	56.0%	63.2%
Semantic	66.0%	64.4%	66.0%	64.3%	52.0%	54.3%
ListLookup	72.0%	73.7%	66.0%	64.9%	64.0%	63.9%
Document	72.0%	70.9%	69.0%	68.2%	74.0%	73.4%
ErrorBased	62.0%	68.0%	62.0%	68.0%	62.0%	68.0%
Readability	54.0%	51.1%	54.0%	47.8%	50.0%	50.2%
AIFeedback	52.0%	50.9%	50.0%	39.8%	45.0%	30.1%
TextVector	**79.0%**	**78.2%**	75.0%	71.0%	69.0%	65.1%
All	77.0%	77.6%	71.0%	69.8%	72.0%	71.9%

Table 14 Results for *advanced EN AI-generated* Wikipedia texts: XGBoost vs. RF vs. MLP ($Acc_{GPTZero}$=79.0%, $F1_{GPTZero}$ =82.7%)

	XGBoost		RF		MLP	
Category	Acc	F1	Acc	F1	Acc	F1
Perplexity	83.0%	82.2%	85.0%	83.8%	83.0%	82.6%
Semantic	75.0%	71.1%	76.0%	75.1%	73.0%	70.2%
ListLookup	83.0%	84.8%	82.0%	82.6%	73.0%	73.2%
Document	90.0%	90.7%	91.0%	91.8%	92.0%	91.8%
ErrorBased	62.0%	71.7%	62.0%	71.7%	59.0%	67.8%
Readability	60.0%	59.7%	59.0%	56.8%	65.0%	63.2%
AIFeedback	66.0%	71.1%	66.0%	71.1%	66.0%	71.1%
TextVector	90.0%	89.1%	**97.0%**	**96.9%**	75.0%	73.8%
All	93.0%	94.0%	95.0%	95.9%	84.0%	82.5%

Table 15 Results for *advanced AI-rephrased EN* Wikipedia texts: XGBoost vs. RF vs. MLP ($Acc_{GPTZero}$=52.0%, $F1_{GPTZero}$ =45.8%)

	XGBoost		RF		MLP	
Category	Acc	F1	Acc	F1	Acc	F1
Perplexity	66.0%	65.6%	65.0%	65.8%	60.0%	63.7%
Semantic	55.0%	56.3%	63.0%	61.5%	61.0%	63.6%
ListLookup	76.0%	75.5%	75.0%	75.3%	72.0%	70.4%
Document	76.0%	74.9%	76.0%	76.2%	77.0%	77.5%
ErrorBased	62.0%	71.7%	62.0%	71.7%	55.0%	62.2%
Readability	58.0%	55.0%	67.0%	66.0%	67.0%	68.0%
AIFeedback	58.0%	61.7%	58.0%	61.7%	58.0%	61.7%
TextVector	71.0%	66.5%	78.0%	75.0%	73.0%	71.3%
All	**82.0%**	**81.7%**	76.0%	76.3%	77.0%	75.2%

The underlying assumption was that this *advanced* way of prompting makes it more difficult to recognize *AI-generated* and *AI-rephrased* texts.

Table 14 shows that the results of our *advanced text generation* detection systems are almost as good as those of our *basic text generation* detection systems which demonstrates that the detection of the *advanced AI-generated* text is not a major challenge for our features. The best performing feature categories are *TextVector* (*Acc*=97.0%, *F1*=96.9%), *Document* (*Acc*=92.0%, *F1*=91.8%), *Perplexity* (*Acc*=85.0%, *F1*=83.8%), and *ListLookup* (*Acc*=83.0%, *F1*=84.8%). Again, among XGBoost, RF, and MLP, no classifier outperforms the other classifiers across all feature categories. Some systems are better than GPTZero ($Acc_{GPTZero}$=79.0%, $F1_{GPTZero}$=82.7%). Our best system *TextVector* (*Acc*=97.0%, *F1*=96.9%) outperforms GPTZero by 22.8% relative in *Acc* and 17.2% relative in *F1*.

In Table 15, we summarize the results of the *advanced text rephrasing* detection systems. The results indicate that these systems perform worse than the *advanced text generation* detection systems but still slightly better than the *basic text rephrasing* detection systems. This demonstrates that detecting *advanced AI-rephrased* with our features is possible, even though more difficult than the detection of *advanced AI-generated* text. The feature categories which perform best are *TextVector* (*Acc*=78.0%, *F1*=75.0%), *ListLookup* (*Acc*=76.0%, *F1*=75.5%) and *Document* (*Acc*=77.0%, *F1*=77.5%). Across our feature categories, none of the classifiers XGBoost, RF, and MLP consistently outperforms the others. As in the previous results, all our systems are better than GPTZero ($Acc_{GPTZero}$=52.0%, $F1_{GPTZero}$=45.8%). Our best performing system *All* (*Acc*=82.0%, *F1*=81.7%) outperforms GPTZero by 57.7% relatively in *Acc* and by an even larger margin of 78.4% relatively in *F1*.

6.2 Classification of human- and AI-generated texts for different languages

In the following sections, we will analyze how our classifiers perform in different languages. Our analyses focus on the following languages: *EN*, *FR*, *DE*, and *ES*.

As visualized in Fig. 12 using the example of $sentiment_{subjectivity}$, the distribution of feature values differs not only depending on the type of production, i.e. if the text is *human-generated*, *AI-generated* or *AI-rephrased* but also substantially depending on the language. The $sentiment_{subjectivity}$ values reflect the degree of objectivity (low values) or subjectivity (high values) of a text. Average $sentiment_{subjectivity}$ values tend to be higher for *AI-generated* text than for *human-generated* and *AI-rephrased* text. Our

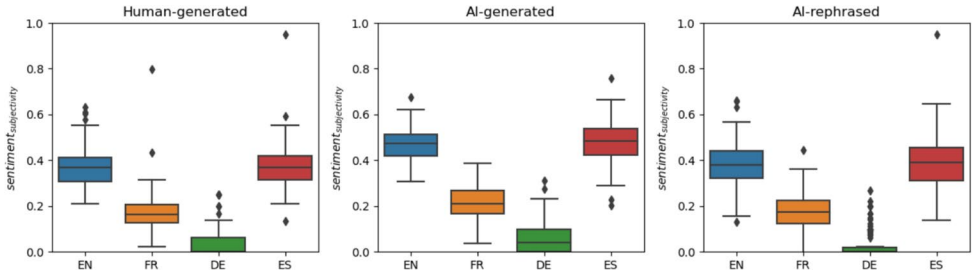

Fig. 12 Distribution of $sentiment_{subjectivity}$ for *EN*, *FR*, *DE*, and *ES*

analysis shows, that *DE* texts are most objective—be it *human-* or *AI-generated*—while *EN* and *ES* are more subjective. Additionally, *AI-generated* texts tend to be more subjective than *AI-rephrased* texts for our languages.

Table 16 summarizes *Acc* and *F1* for identifying texts that were *AI-generated* and those that were *AI-rephrased* in *EN*, *FR*, *DE*, and *ES* using the *basic prompting*. The classifiers for the detection of *AI-generated* text were more effective than those for *AI-rephrased* texts in each of the languages. In the following sections, we will have a detailed look at the results for the respective languages.

6.2.1 English

In the next paragraphs, we will present the performances of our *EN text generation* and *rephrasing* detection systems.

Text generation detection systems

As described in Sect. 6.1 and presented in Table 16 in comparison to the other languages, the results for *EN* show that the RF model which uses *All* features achieves the highest performance (*Acc*=98.0%, *F1*=98.0%). Following closely is the RF model using *TextVector* features with a performance of *Acc*=95.0%, and *F1*=94.9%. Moreover, the RF model using the *Document* features demonstrates comparable results (*Acc*=92.0%, *F1*=92.6%). XGBoost utilizing the *ErrorBased* features yields the lowest performance among our evaluated systems (*Acc*=55.0%, *F1*=61.7%).

A comparison with GPTZero ($Acc_{GPTZero}$=76.0%, $F1_{GPTZero}$=78.9%) indicates the high performance across most of our *EN* models. Our best model combining *All* features surpasses GPTZero by 28.9% relatively in terms of *Acc* and 24.2% relatively in terms of *F1*. Additionally, ZeroGPT achieves an accuracy of 78.0% ($Acc_{ZeroGPT}$) and an *F1* of 81.8% ($F1_{ZeroGPT}$). Our optimal model demonstrates a 25.6% relative improvement in accuracy and a 19.8% relative improvement in *F1* compared to ZeroGPT.

Text rephrasing detection systems

As described in Sect. 6.1 and presented in Table 16 in comparison to the other languages, the performances for the *EN AI-rephrased* text detection are worse than the *AI-generated* text detection for all feature categories except *ErrorBased* (*Acc*=62.0%, *F1*=68.0%). The best-performing system is the XGBoost system using *TextVector* features (*Acc*=79.0%, *F1*=78.2%), followed by the MLP system using *Document* features (*Acc*=78.0%, *F1*=76.1%). The MLP system utilizing the *AIFeedback* feature performs worst. All our *EN text rephrasing* detection systems were able to outperform GPTZero ($Acc_{GPTZero}$=43.0% and $F1_{GPTZero}$=27.8%). Our best-performing *TextVector* feature system even surpasses GPTZero by 83.7% relatively in *Acc* and even 159.8% relatively in *F1*. ZeroGPT achieves 49.0% $Acc_{ZeroGPT}$ and 43.9% $F1_{ZeroGPT}$. Thus, *Document* features outperform it by 61.2% relatively in *Acc* and 81.5% relatively in *F1*.

6.2.2 French

In the next paragraphs, we will present the performances of our *FR text generation* and *rephrasing* detection systems.

Text generation detection systems

The results for *FR* in Table 16 reveal that for the detection of *AI-generated* text, the system combining *All* features in an RF achieves the highest performance (*Acc*=95.0%, *F1*=95.0%). The second-best system is the XGBoost system utilizing *Document* features (*Acc*=94.0%, *F1*=94.2%), followed by the XGBoost system using *TextVector* features (*Acc*=94.0%, *F1*=94.1%). The least effective systems are those based on the *AIFeedback* feature. Our best performing *FR* system with *All* features surpasses ZeroGPT ($Acc_{ZeroGPT}$ =62.0%, $F1_{ZeroGPT}$=72.6%) by 53.2% relatively in *Acc* and 30.9% relatively in *F1*.

Text rephrasing detection systems

The performances of the systems to detect *FR AI-rephrased* texts are worse than the performances of the system to detect *FR AI-generated* texts across all feature categories, except for the MLP system which uses the *AIFeedback* feature (*Acc*=55.0%, *F1*=53.4%). The best-performing system is the one that integrates *All* features in an XGBoost model (*Acc*=89.0%, *F1*=87.9%), followed by the XGBoost system employing *Document* features (*Acc*=86.0%, *F1*=85.3%) and the XGBoost system utilizing *TextVector* features (*Acc*=77.0%, *F1*=77.3%). The least effective systems are again those based on the *AIFeedback* feature. Our best *FR* system with *All* features outperforms ZeroGPT ($Acc_{ZeroGPT}$=57.0%, $F1_{ZeroGPT}$=67.4%) by 56.1% relatively in *Acc* and 30.4% relatively in *F1*.

Table 16 Results for the detection of basic *EN*, *FR*, *DE* and *ES AI-generated* and *AI-rephrased* texts

		Generated						Rephrased					
		XGBoost		RF		MLP		XGBoost		RF		MLP	
Category	Language	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Perplexity	EN	83.0	82.2	87.0	85.3	82.0	82.1	52.0	48.7	55.0	54.6	56.0	63.2
	FR	62.0	60.3	69.0	66.8	68.0	69.0	50.0	50.2	53.0	44.2	56.0	58.8
	DE	74.0	74.0	76.0	76.1	81.0	80.6	53.0	53.6	61.0	60.4	56.0	62.7
	ES	82.0	82.3	83.0	82.4	82.0	83.6	56.0	55.4	63.0	63.7	62.0	67.3
Semantic	EN	72.0	72.9	75.0	75.6	73.0	72.3	66.0	64.4	66.0	64.3	52.0	54.3
	FR	61.0	55.8	67.0	65.6	63.0	59.4	55.0	48.2	57.0	50.0	51.0	52.9
	DE	64.0	58.3	64.0	59.8	63.0	63.3	56.0	59.9	54.0	54.4	62.0	60.1
	ES	72.0	69.9	75.0	73.8	76.0	75.7	58.0	56.1	58.0	52.4	53.0	56.3
ListLookup	EN	72.0	72.1	79.0	78.5	71.0	67.8	72.0	73.9	67.0	67.5	69.0	70.3
	FR	72.0	73.0	76.0	76.7	67.0	62.9	66.0	62.6	65.0	65.5	64.0	63.2
	DE	74.0	75.8	79.0	77.8	72.0	74.1	57.0	59.1	58.0	59.2	50.0	52.0
	ES	78.0	79.6	82.0	84.1	73.0	76.8	75.0	75.2	80.0	81.3	77.0	78.4
Document	EN	91.0	91.6	92.0	92.6	87.0	86.0	70.0	69.6	71.0	70.8	78.0	76.1
	FR	94.0	94.2	91.0	90.8	92.0	92.2	86.0	85.3	84.0	80.8	81.0	81.2
	DE	87.0	87.2	90.0	89.6	88.0	88.0	**72.0**	**71.9**	67.0	66.7	71.0	71.3
	ES	96.0	96.2	98.0	98.1	87.0	88.5	84.0	83.4	83.0	82.0	**86.0**	**86.4**
ErrorBased	EN	55.0	61.7	55.0	61.7	56.0	63.9	62.0	68.0	62.0	68.0	62.0	68.0
	FR	62.0	64.2	63.0	67.2	61.0	65.5	53.0	56.0	56.0	58.9	56.0	59.7
	DE	67.0	67.1	67.0	67.1	67.0	69.8	62.0	61.9	62.0	63.5	56.0	50.7
	ES	70.0	71.2	71.0	71.9	71.0	74.6	59.0	56.8	61.0	56.3	64.0	65.2
Readability	EN	60.0	56.3	63.0	59.3	60.0	56.8	54.0	51.1	54.0	47.8	50.0	50.2
	FR	61.0	64.7	62.0	66.0	65.0	67.4	59.0	58.3	60.0	60.6	52.0	31.6
	DE	57.0	53.5	53.0	51.5	57.0	53.6	48.0	41.9	45.0	39.1	45.0	44.9
	ES	74.0	73.7	74.0	72.1	69.0	66.6	54.0	49.1	61.0	50.7	56.0	52.5
AIFeedback	EN	62.0	67.1	62.0	67.1	62.0	68.1	52.0	50.9	50.0	39.8	45.0	30.1
	FR	52.0	24.2	52.0	24.2	48.0	37.2	42.0	33.6	42.0	33.6	55.0	53.4
	DE	49.0	46.1	47.0	35.0	50.0	43.4	52.0	61.8	52.0	61.8	50.0	54.3
	ES	52.0	7.3	52.0	7.3	52.0	20.6	50.0	0.0	52.0	7.3	49.0	25.7
TextVector	EN	90.0	89.9	95.0	94.9	83.0	81.7	**79.0**	**78.2**	75.0	71.0	69.0	65.1
	FR	94.0	94.1	93.0	93.0	85.0	85.4	77.0	77.3	75.0	75.2	68.0	64.2
	DE	87.0	87.0	94.0	94.0	90.0	90.8	68.0	67.5	72.0	67.3	72.0	71.7
	ES	84.0	84.5	91.0	89.5	81.0	76.6	76.0	74.0	76.0	73.6	68.0	64.4
All	EN	90.0	90.9	**98.0**	**98.0**	87.0	87.8	77.0	77.6	71.0	69.8	72.0	71.9
	FR	94.0	94.4	**95.0**	**95.0**	88.0	89.2	**89.0**	**87.9**	86.0	84.2	74.0	66.4
	DE	94.0	93.8	**97.0**	**96.9**	87.0	86.6	70.0	71.6	71.0	68.3	70.0	71.6
	ES	94.0	94.4	**99.0**	**99.0**	90.0	90.2	83.0	82.2	83.0	82.9	78.0	76.1

6.2.3 German

In the next paragraphs, we will demonstrate the performances of our *DE text generation* and *rephrasing* detection systems.

Text generation detection systems

The results for *DE* in Table 16 indicate that for the detection of *AI-generated* text the system using *All* features in an RF model performs best (*Acc*= 97.0%, *F1*= 96.9%). The second-best system is the RF system utilizing *TextVector* features (*Acc*= 94.0%, *F1*= 94.0%), followed by the RF system employing *Document* features (*Acc*= 90.0%, *F1*=89.6%). Similar to previous languages, the least successful setups are those using the *AIFeedback* feature. Our best-performing *FR* system with *All* features outperforms ZeroGPT ($Acc_{ZeroGPT}$= 65.0%, $F1_{ZeroGPT}$= 70.9%) by 49.2% relatively in *Acc* and 36.7% relatively in *F1*.

Text rephrasing detection systems

The performances of the *DE text rephrasing* detection systems perform worse than those of the *DE text generation*

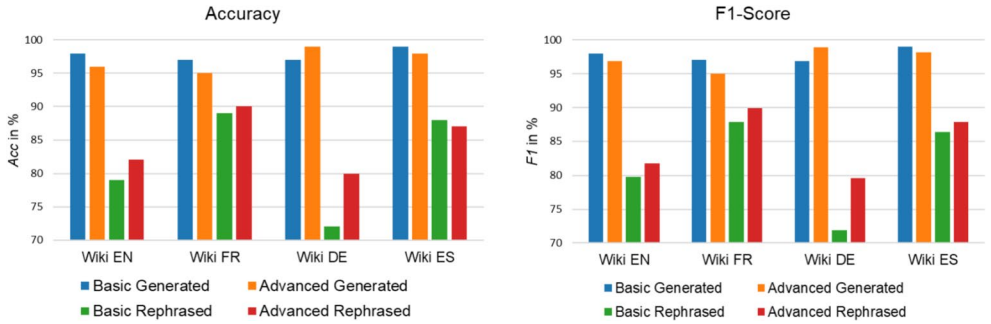

Fig. 13 Comparison of *Acc* and *F1* across the different languages

detection systems across all feature categories, except for systems utilizing the *AIFeedback* features. The best-performing system is the XGBoost model utilizing *Document* features (*Acc*= 72.0%, *F1*= 71.9%), followed by the MLP system utilizing *TextVector* features (*Acc*= 72.0%, *F1*= 71.7%). The least effective systems are those utilizing the *Readability* feature. Our top *DE* system with *Document* features outperforms ZeroGPT ($Acc_{ZeroGPT}$=48.0%, $F1_{ZeroGPT}$ =49.5%) by 45.5% relatively in *Acc* and 45.3% relatively in *F1*.

6.2.4 Spanish

In the next paragraphs, we will present the performances of our *ES text generation* and *rephrasing* detection systems.

Text generation detection systems

The outcomes for *ES* in Table 16 indicate that the system integrating *All* features in an RF model achieves the highest performance (*Acc*= 99.0%, *F1*= 99.0%). The second-best configuration is the RF system which is based on *Document* features (*Acc*=98.0%, *F1*=98.1%), followed by the RF system which uses *TextVector* features (*Acc*= 91.0%, *F1*= 89.5%) and the RF system based on *ListLookup* features (*Acc*= 82.0%, *F1*= 84.1%). Consistent with the previously analyzed languages, the least effective setups are those employing the *AIFeedback* feature. The *F1* of 7.3% is notably low due to the feature classifying the text as *AI-generated* text in almost all cases. Our top *ES* system with *All* features surpasses ZeroGPT ($Acc_{ZeroGPT}$=60.0%, $F1_{ZeroGPT}$ =71.5%) by 65.0% relatively in *Acc* and 38.5% relatively in *F1*.

Text rephrasing detection systems

The performances of the systems to detect *ES AI-rephrased* texts are worse than those of the systems to detect *ES AI-generated* texts across all feature categories. The best-performing system is the RF system based on *Document* features (*Acc*= 86.0%, *F1*= 86.4%). The second-best system is the system combining *All* features in an RF model (*Acc*= 83.0%, *F1*=82.9%), followed by the RF system utilizing *ListLookup* features (*Acc*= 80.0%, *F1*= 81.3%). The least effective systems are those based on the *AIFeedback* feature. The *F1* of 0% and 7.3% are so low as the classifier based on this feature classifies the text as *AI-generated* text in almost all cases. Our top *ES* system with *Document* features outperforms ZeroGPT ($Acc_{ZeroGPT}$=52.0%, $F1_{ZeroGPT}$=63.7%) by 65.4% relatively in *Acc* and 25.6% relatively in *F1*.

6.2.5 Comparing performance across languages

Figure 13 summarizes *Acc* and *F1* of the best classifiers for the four languages. As the Figure and the results in Table 16 show, the best performances for the detection of *AI-generated* text are achieved through the integration of *All* features. Analyzing the systems employing all features, the *Acc* for the *AI-generated FR* and *DE* texts is similar with 97.0%, while for the *AI-generated EN* texts, it is slightly better with 98.0%. The highest *F1* for the *AI-generated DE* text classifier is 96.9%, which is only slightly worse than the classifiers trained on our *EN* and *FR* texts, achieving 98.0% and 97.1%, respectively. The best classifier trained on the *AI-generated ES* texts exhibits a performance of 99.0% *Acc* and 99.0% *F1*. Comparing the system performance trained on the *AI-generated* texts shows that the performances of the respective classifiers across languages are closely together.

For *AI-rephrased* texts, the performances of the systems using *All* features demonstrate more variability across the languages. While the *EN* classifier achieves 79.0% *Acc* on the *AI-rephrased* texts, the best *FR* classifier has an *Acc* of 89.0% on the *AI-rephrased* texts. The *AI-rephrased* detection system for *DE* only achieves 72.0% *Acc*. Compared to the best system to detect *DE AI-rephrased* text, the *FR* system demonstrates a 23.6% relative improvement in *Acc*. The *Acc* for the system to detect *ES AI-rephrased* text is 1% worse than the *FR* system. For *F1*, similar conclusions can be drawn across the languages. Thus, our examined features yield different performances for the detection of *AI-rephrased* texts across the evaluated languages.

Table 17 Results of our *basic AI-generated* and *AI-rephrased* news articles detection

	AI-Generated						AI-Rephrased					
	XGBoost		RF		MLP		XGBoost		RF		MLP	
Category	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
Perplexity	85.0	85.8	90.0	90.1	82.0	80.3	81.0	80.5	80.0	77.9	75.0	76.9
Semantic	55.0	52.5	60.0	60.6	59.0	57.1	60.0	53.8	62.0	56.0	59.0	63.6
ListLookup	75.0	77.3	70.0	73.5	78.0	80.0	91.0	91.2	94.0	94.3	85.0	86.8
Document	85.0	85.5	87.0	86.4	82.0	82.7	93.0	92.8	**96.0**	**96.1**	92.0	92.4
ErrorBased	77.0	77.2	79.0	77.8	73.0	72.7	76.0	76.3	75.0	75.5	75.0	77.5
Readability	53.0	55.1	62.0	62.1	60.0	54.5	60.0	60.4	59.0	59.9	57.0	26.7
AIFeedback	50.0	0.0	50.0	0.0	50.0	26.7	50.0	0.0	50.0	0.0	50.0	26.7
TextVector	87.0	86.2	92.0	91.9	91.0	90.9	90.0	90.4	94.0	94.2	87.0	86.1
All	**94.0**	**93.9**	91.0	91.8	87.0	87.1	95.0	95.0	95.0	95.1	92.0	90.9

6.3 Classification of human- and AI-generated texts for the news domain

Writing a text for the educational domain shows stylistic differences compared to other domains. The classifiers we have described so far were all based on Wikipedia texts as these compare to texts from the educational domain. To analyze if our developed features can also be used in other domains, we trained and evaluated systems with the *AI-generated* and *AI-rephrased* news texts from our *Multilingual Human-AI-Generated Text Corpus*. Similar to our previous analyses, we compared the classification of texts created with *basic prompting* and *advanced prompting* for *AI-generated* and *AI-rephrased* texts.

6.3.1 Basic prompting

Table 17 indicates that the best *news text generation* detection system is the system that combines *All* features in an XGBoost (*Acc*= 94.0%, *F1*= 93.9%). The 2nd-best system is the RF system that uses *TextVector* features (*Acc*= 92.0%, *F1*= 91.9%), followed by the MLP system that also uses *TextVector* features (*Acc*= 91.0%, *F1*= 90.9%). The worst-performing systems are the XGBoost and the RF systems that use the *AIFeedback* features (*Acc*= 50.0%, *F1*= 0.0%). Compared to GPTZero ($Acc_{GPTZero}$=68.0%, $F1_{GPTZero}$=74.6%), most of our systems perform better. Our best system with *All* features outperforms GPTZero by 38.2% relative in *Acc* and 25.9% relative in *F1*.

The best *news text rephrasing* detection system is the system that employs our *Document* features in an RF (*Acc*= 96.0%, *F1*= 96.1%). The second-best system is the system that combines *All* features in an RF (*Acc*= 95.0%, *F1*= 95.1%), followed by the RF system that uses *ListLookup* features (*Acc*= 94.0%, *F1*= 94.3%). The worst-performing systems are the XGBoost and the RF systems that use the *AIFeedback* features (*Acc*= 50.0%, *F1*= 0.0%). The best system that combines our *Document* features in an RF outperforms GPTZero ($Acc_{GPTZero}$=41.0%, $F1_{GPTZero}$=40.0%) by 134.1% relative in *Acc* and 140.3% relative in *F1*.

6.3.2 Advanced prompting

Table 18 illustrates that the best system for the *AI-generated* news articles based on *advanced prompting* is the system that combines *All* features in an RF (*Acc*= 92.0%, *F1*= 91.7%). Similar to the basic prompting, our 2nd-best system is the RF system that uses *TextVector* features (*Acc*= 92.0%, *F1*= 91.6%), followed by an XGBoost system that uses *All* features (*Acc*= 91.0%, *F1*= 90.9%). The worst-performing systems are the XGBoost and the RF systems that use the *AIFeedback* features (*Acc*= 50.0%, *F1*= 0.0%). Compared to GPTZero ($Acc_{GPTZero}$=68.0%, $F1_{GPTZero}$=74.6%), most of our systems perform better. Our best system with *All* features outperforms GPTZero by 35.3% relative in *Acc* and 22.9% relative in *F1*.

As indicated by the results shown in Table 18, detecting *AI-rephrased* news articles created with *advanced prompting* performs better than the detection of *AI-generated* news articles. The best system is the system that combines *All* features in an RF (*Acc*= 96.0%, *F1*= 96.0%). The 2nd-best system is the system based on our *Document* features in an XGBoost (*Acc*= 96.0%, *F1*= 95.8%), followed by the XGBoost system that uses *ListLookup* features (*Acc*= 94.0%, *F1*= 94.2%). The worst-performing systems are the XGBoost and the RF systems that use the *AIFeedback* features (*Acc*= 50.0%, *F1*= 0.0%). The best system that combines *All* features in an RF outperforms GPTZero ($Acc_{GPTZero}$=38.0%, $F1_{GPTZero}$ =34.8%) by 152.6% relative in *Acc* and 175.9% relative in *F1*.

Table 18 Results of our *advanced AI-generated* and *AI-rephrased* news articles detection

	AI-Generated						AI-Rephrased					
	XGBoost		RF		MLP		XGBoost		RF		MLP	
Category	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
Perplexity	90.0	90.1	91.0	90.8	87.0	88.0	80.0	79.7	79.0	78.3	71.0	73.5
Semantic	53.0	50.6	58.0	55.3	58.0	60.2	63.0	58.1	64.0	61.6	54.0	57.4
ListLookup	73.0	73.6	69.0	70.3	65.0	66.0	94.0	94.2	94.0	94.2	91.0	91.9
Document	85.0	85.8	87.0	87.3	85.0	84.9	96.0	95.8	95.0	94.9	88.0	87.7
ErrorBased	81.0	80.5	81.0	80.5	73.0	75.9	79.0	80.6	81.0	82.0	76.0	77.9
Readability	60.0	63.5	65.0	66.9	61.0	64.3	55.0	55.9	61.0	60.1	48.0	49.6
AIFeedback	50.0	0.0	50.0	0.0	50.0	40.0	50.0	0.0	50.0	0.0	50.0	26.7
TextVector	90.0	90.0	92.0	91.6	82.0	79.6	93.0	92.9	94.0	94.1	83.0	83.3
All	92.0	90.9	**92.0**	**91.7**	92.0	89.9	92.0	92.4	**96.0**	**96.0**	88.0	87.3

7 Conclusion and future work

In the presented study, we investigated features for detecting *AI-generated* and *AI-rephrased* text. To provide texts that are written similar to essays or homework in education, we created a new text corpus—the *Multilingual Human-AI-Generated Text Corpus*—based on Wikipedia texts covering 10 categories from the educational domain: *Biology*, *chemistry*, *geography*, *history*, *IT*, *music*, *politics*, *religion*, *sports*, and *visual arts*. These texts were generated in four different languages: *EN*, *FR*, *DE*, and *ES*. To test if our features can also be successfully used in another domain, we extended the text corpus by *EN AI-generated* and *AI-rephrased* news articles from five different categories: *Crime*, *entertainment*, *politics*, *science*, and *sports*. Being able to detect *AI-generated* or *AI-rephrased* news articles can, for instance, be used as an additional feature for fake news detection.

Concerning the *EN* Wikipedia texts, we achieve an *F1* of 98.0% when classifying *basic human-generated/AI-generated* texts, and an *F1* of 78.9% for *basic human-generated/AI-rephrased* texts. Additionally, we obtain an *F1* of 96.9% for classifying *human-generated/AI-generated* text created with *advanced prompting* and an *F1* of 81.7% for *human-generated/AI-rephrased* text with *advanced prompting*.

In addition to *EN*, we analyzed the performances of our features for the classification of *AI-generated* and *AI-rephrased* texts for *FR*, *DE*, and *ES*. For *AI-generated* texts, we achieve the best performance, when we combine all features: All *F1* scores are closely together with 99% for *ES*, 98% for *EN*, 96.9% for *DE* and 95% for *FR*. This indicates that using our features for the detection of *AI-generated* text results in very good performances across our analyzed languages and there may be a potential that the same features can also be used for other languages. For the detection of *AI-rephrased* texts, the systems with *All* features outperform systems with other features in many cases. For *FR* we achieve an *F1* of 87.9%. For *DE* (*F1*=71.9%) and *ES* (*F1*=86.4%) we achieve the best results using only *Document* features while for *EN* the *TextVector* features yield the best result (*F1*=78.2%).

Our best *news text generation* detection system uses *All* features and achieves an *F1* of 93%. Our best *news text rephrasing* detection system obtains an *F1* of 96%. The best system for the *AI-generated* news articles based on *advanced prompting* combines *All* features and obtains an *F1* of 91.7%. Detecting *AI-rephrased* news articles created with *advanced prompting* performs better than the detection of AI-generated news articles: The best system is the system that combines *All* features and obtains an *F1* of 96.0%.

Our best systems from all analyzed languages and domains outperform GPTZero and ZeroGPT by far. Our best *basic prompting text rephrasing* detection system even outperforms GPTZero by 181.3% relative to *F1*. Thus, our research indicates that using our features can substantially improve the detection performance of *AI-generated* and *AI-rephrased* texts.

Knowledge about the features and methods that can be used to detect *AI-generated* texts can be exploited to create AI-generated content that bypasses detection systems. In this way, people could abuse the ability to create high-quality *AI-generated* text content, for instance, to spread misinformation more effectively. For this reason, it is important to further improve our system. In future work, we therefore plan to extend our research toward other languages and domains. In particular, we plan to investigate whether our features can be transferred to languages with different linguistic characteristics such as Chinese, or Arabic. Expanding our dataset size and including additional domains such as scientific literature, or social media posts can improve generalizability. Moreover—besides *basic prompting* and *advanced prompting*—further prompting variants and their impact on the classification performance will be evaluated.

As LLM-based chatbots continue to evolve, we additionally plan to evaluate texts generated by other LLM-based chatbots such as Gemini, Llama 2, and future ChatGPT versions. Additionally, further features could be analyzed including an analysis of their importance based on the respective domains and languages. Depending on the text domain, detailed error analysis can help to better understand false negatives and false positives and allow further model fine-tuning depending on how critical the errors are in the respective domains. Further hyperparameter tuning and the evaluation of other classifiers can also help to improve performance. By addressing these aspects, we aim to enhance the robustness, accuracy, and generalizability of our findings in classifying *human-* and *AI-generated* texts.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability To contribute to the improvement of the detection of AI-generated text, we share our corpus with the research community: https://github.com/iu-ai-research/human-AI-generatedTextCorpus.

Declarations

Conflict of interest All authors have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Adiwardana, D., *et al.* (2020). Towards a human-like open-domain Chatbot. ArXiv Preprint arXiv:2001.09977.

Arteaga, D., Arenas, J., Paz, F., Tupia, M., & Bruzza, M. (2019). *Design of information system architecture for the recommendation of tourist sites in the city of Manta, Ecuador through a chatbot*, (pp. 1–6). IEEE.

Baidoo-Anu, D., & Owusu Ansah, L. (2023). Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Available at SSRN 4337484*.

Bird, S., & Loper, E. (2004). *NLTK: The natural language toolkit*, (pp. 214–217). Association for Computational Linguistics. https://aclanthology.org/P04-3031.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Brown, T. B., Mann, B., Ryder, N.,...., & Amodei, D. (2020). Language models are few-shot learners. *CoRR* arXiv:abs/2005.14165.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system, KDD '16, (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Components. Components (2023). https://components.one.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, (pp. 4171–4186).

Dibitonto, M., Leszczynska, K., Tazzi, F., & Medaglia, C. M. (2018). *Chatbot in a campus environment: Design of LiSA, a virtual assistant to help students in their university life,* (pp.103–116). Springer.

Ethnologue. (2023). What are the top 200 most spoken languages? https://www.ethnologue.com/insights/ethnologue200.

Falala-Séchet, C., Antoine, L., Thiriez, I., & Bungener, C. (2019). *OWLIE: A Chatbot that provides emotional support for coping with psychological difficulties*, (pp. 236–237).

Flesch, R. F. (1948). A new readability yardstick. *The Journal of Applied Psychology, 32*(3), 221–233.

Gehrmann, S., Strobelt, H., & Rush, A. (2019). *GLTR: Statistical detection and visualization of generated text,* (pp.111–116). Association for Computational Linguistics.

Guo, B., Zhang, X.,Wang, Z., Jiang, M., Nie, J.,Ding, Y., Yue, J., & Wu, Y. (2023). How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. arXiv:2301.07597.

Jeblick, K., Schachtner, B., Dexl, J., Mittermeier, A., Stvüber, A. T., Topalis, J., Weber, T., Wesp, P., Sabel, B. O., Ricke, J., & Ingrisch, M. (2023). ChatGPT makes medicine easy to swallow: An exploratory case study on simplified radiology reports. *European Radiology, 34*, 2817–2825.

Jiao, W., Wang, W., Huang, J.-t., Wang, X., & Tu, Z. (2023). Is ChatGPT a good translator? A preliminary study. ArXiv Preprint arXiv:2301.08745.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computer Survey*. https://doi.org/10.1145/3571730

Kincaid, J. P., Fishburne Jr., R. P., Rogers, R. L., & Chissom, B. S. (1975). *Derivation of new readability formulas (Automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*.

Kumarage, T., Garland, J., Bhattacharjee, A., Trapeznikov, K., Ruston, S., & Liu, H. (2023). Stylometric detection of AI-generated text in Twitter timelines. arXiv:2303.03697.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *CoRR* arXiv:abs/1907.11692.

Mesko, B. (2023). The ChatGPT (generative artificial intelligence) revolution has made artificial intelligence approachable for medical professionals. *Journal of Medical Internet Research, 25*, e48392.

Mindner, L., Schlippe, T., Schaaff, K. Schlippe, T., Cheng, E. C. K., & Wang, T. (eds) (2023). Classification of human- and AI-generated texts: Investigating features for ChatGPT. In Schlippe, T., Cheng, E. C. K. & Wang, T. (Eds.) *Artificial intelligence in education technologies: New development and innovative practices*, (pp. 152–170. Springer Nature.

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). DetectGPT: Zero-shot machine-generated text detection using probability curvature.

Mitrović, S., Andreoletti, D., & Ayoub, O. (2023). ChatGPT or human? Detect and explain. Explaining decisions of machine learning model for detecting short ChatGPT-generated text. arXiv preprint arXiv:2301.13852.

Mooney, P. (2022). Kaggle machine learning and data science survey 2022. https://kaggle.com/competitions/kaggle-survey-2022.

Murtagh, F. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing, 2*, 183–197.

Natalie. (2023). What is ChatGPT? https://help.openai.com/en/articles/6783457-what-is-chatgpt.

Pelau, C., Dabija, D.-C., & Ene, I. (2021). What makes an AI device human-like? The role of interaction quality, empathy and perceived psychological anthropomorphic characteristics in the acceptance of artificial intelligence in the service industry. *Computers in Human Behavior, 122*, 106855.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*, (pp. 3982–3992). Association for Computational Linguistics. https://aclanthology.org/D19-1410.

Roberts, A., Raffel, C., Lee, K., Matena, M., Shazee, N., Liu, P. J., Narang, S., Li, W., & Zhou, Y. (2019). *Exploring the limits of transfer learning with a unified text-to-text transformer*. Google: Tech. Rep.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: Smaller, cheaper and lighter: Faster*.

Schaaff, K., Reinig, C., & Schlippe, T. (2023). Exploring ChatGPT's empathic abilities. In *2023 11th international conference on affective computing and intelligent interaction (ACII 2023)* (pp. 1–8). IEEE Computer Society. https://doi.ieeecomputersociety.org/10.1109/ACII59096.2023.10388208.

Schaaff, K., Schlippe, T., Mindner, L. Abbas, M., & Freihat, A. A. (eds) (2023). Classification of human- and AI-generated texts for English, French, German, and Spanish. In Abbas, M. & Freihat, A. A. (Eds.) *The 6th international conference on natural language and speech processing (ICNLSP 2023)*, (pp. 1–10). Association for Computational Linguistics, Online. https://aclanthology.org/2023.icnlsp-1.1.

Shijaku, R., & Canhasi, E. (2023). ChatGPT generated text detection.

Shrivastava, R. (2023). With seed funding secured, AI detection tool GPTZero launches new browser plugin. https://www.forbes.com/sites/rashishrivastava/2023/05/09/with-seed-funding-secured-ai-detection-tool-gptzero-launches-new-browser-plugin.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J.,Radford, A., Krueger, G., Kim, J. W., Kreps, S., McCain, M., Newhouse, A., Blazakis, J., McGuffie, K., & Wang, J. (2019). Release strategies and the social impacts of language models. arXiv:1908.09203.

Soni, M., & Wade, V. (2023). Comparing abstractive summaries generated by ChatGPT to real summaries through blinded reviewers and text classification algorithms. arXiv:2303.17650.

Taecharungroj, V. (2023). "What can ChatGPT do?" Analyzing early reactions to the innovative AI chatbot on Twitter. *Big Data and Cognitive Computing, 7*, 35.

Thompson, P. (2023). A developer built a 'Propaganda machine' using OpenAI Tech to highlight the dangers of mass-produced AI disinformation. https://www.businessinsider.com/developer-creates-ai-disinformation-system-using-openai-2023-9.

Touvron, H., *et al.* (2023). LLaMA: Open and efficient foundation language models. arXiv:2302.13971.

Vu, N. T., Schlippe, T., Kraus, F., & Schultz, T. (2010). *Rapid bootstrapping of five Eastern European languages using the rapid language adaptation toolkit*. https://api.semanticscholar.org/CorpusID:12942559.

Wankhade, M., Rao, A., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review, 55*, 5731–5780.

Yu, P., Chen, J., Feng, X., & Xia, Z. (2023). CHEAT: A large-scale dataset for detecting ChatGPT-writtEn AbsTracts. arXiv:2304.12008.

Zaitsu, W., & Jin, M. (2023). Distinguishing ChatGPT(-3.5, -4)-generated and human-written papers through Japanese stylometric analysis. arXiv:2304.05534.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.