Controllable Text Generation for Style and Content

Michele Papucci ⋆ ◇

⋆ ItaliaNLP LAB @ CNR-ILC

◇ Università di Pisa

michele.papucci@phd.unipi.it

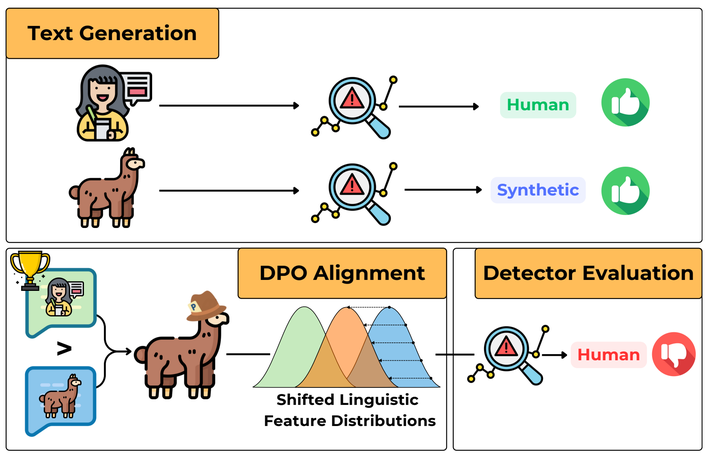
Introduction, Motivation, and Goal

Nowadays, Large Language Models (LLMs) have become the standard state-of-the-art technique to solve most Natural Language Processing (NLP) tasks, especially in text generation tasks. While extremely good at solving those, LLMs still suffer from major problems like following instructions or constraints, generating unfaithful or untrue texts (hallucinations), and their decision-making process being unknown. My project aims at solving these problems, and it's focused on using Controllable Text Generation (CTG) and Interpretability techniques to make the model adhere to specific syntactic or style-related constraints and to detect and prevent the generation of hallucinations.
Here are presented two works that use two different CTG techniques (a Decoding technique and a Reinforcement Learning technique) to control the model's output. In particular, the focus is on constraining the model to produce text that aligns with desired linguistic patterns.

Controllable Text Generation Experiments

Shifting Model Writing Style to Fool Detectors

We defined a CTG pipeline that by fine-tuning language models with Direct Preference Optimization (DPO), shifts the style of MGTs to resemble Human-written Texts (HWTs), making them **harder to detect**.



We present two techniques to build a preference dataset for a DPO fine-tuning:

- *dpo*: We create a parallel dataset of (HWT, MGT), and label the **HWT as the preferred option**.
- *dpo-ling*: We train a Linear SVM to solve the MGT detection task on the text linguistic profiling obtained with ProfilingUD. Then, we take the top-k features that have the highest absolute coefficient for the SVM classification. For each of these features, we select pairs of (HWT, MGT) that **maximize that feature difference**, and tag the HWT as the preferred. The objective is to take the top-most representative couples of sentences for each of the most important features for the SVM detection.

	Performance Drop (F1) on Llama - XSUM			
	Mage	Radar	LLM-DetectAIve	Binoculars
dpo	**0.36**	0.15	**0.19**	**0.66**
dpo-ling	0.29	**0.36**	0.18	0.61

What we found is that the *dpo* approach was the one that maximized the *performance drop* of the four state-of-the-art detectors we tested.
Instead, *dpo-ling* was the technique that performed better in the objective of **aligning feature-specific distribution of the generated texts to the human writing style**.

Human raters' performances were *unaffected* by the DPO fine-tuning, remaining at around 50% accuracy, the same as blindly guessing.

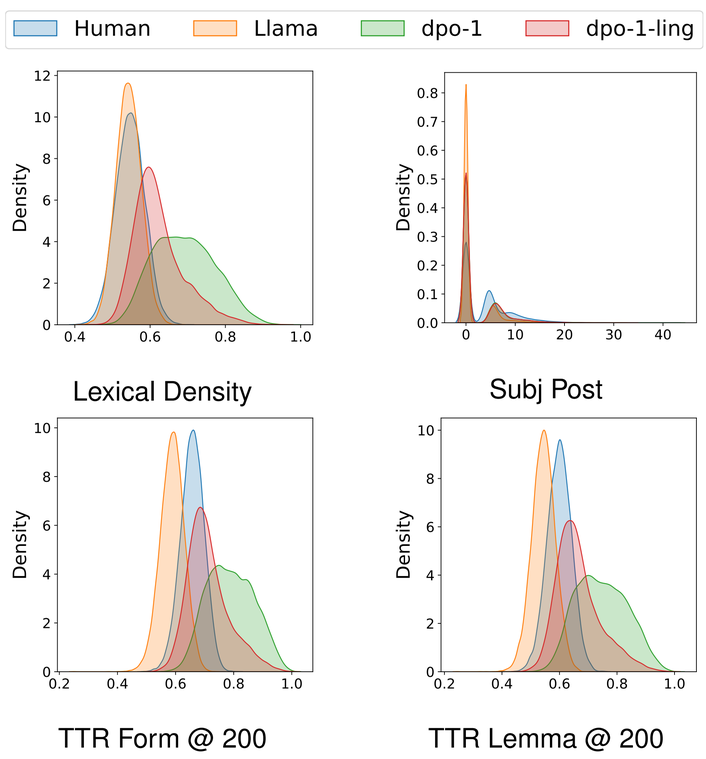


Figure 1: The distribution of selected linguistic features on Llama XSUM generation.

Generating Multi-level Text Simplification

LLMs can be used as a tool for generating datasets for low-resource languages. We propose a pipeline for the creation and evaluation of a Sentence Simplification resource for Italian:

1. We identified the best-performing Italian LLM for Sentence Simplification (LlaMantino-2);
2. We sampled simplifications from the model using original sentences from two different domains (Wikipedia and PaWaC) using the **Diverse Beam Search Decoding technique**, ensuring we obtained 10 different simplifications for each sentence;
3. We evaluated the resulting sentence pairs in terms of their linguistic feature diversity and variation in readability levels.

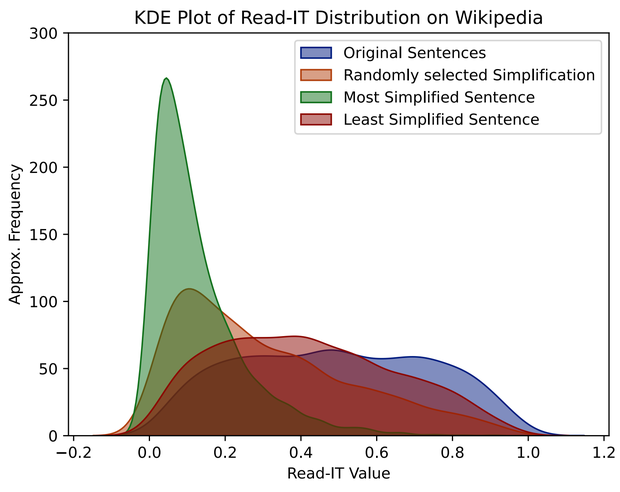
	Wikipedia	**Pawac**
	Pillai's Trace	Pillai's Trace
Original vs Least Simplified	.12	.16
Original vs Randomly-Selected	.18	.19
Original vs Most Simplified	.44	.46

We found that the selected LLM was capable of generating **multiple simplifications**, at **different readability levels** (evaluated with Read-IT), for each original sentence.

By looking at the linguistic profiling of the generated simplifications, we found that the model **used simplification strategies similar to those used in manually crafted simplification resources**. In particular, the two methodologies shared:

- a lower *number of tokens*;
- fewer *pronouns*, *adverbs*, and *punctuation marks*, a higher proportion of *determiners*;
- a *shallower syntactic trees*, and *shorter dependency links*.

With this newly created resource, we aim to train models that can be used to produce **simplification aimed at a specific readability level of the user**.



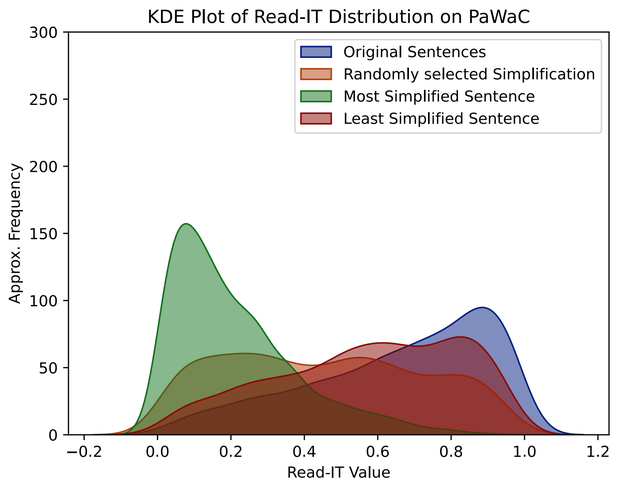


Figure 2: For both Wikipedia on the left and PaWaC on the right, the Kernel Density Estimate for the READ-IT.

We find various linguistic patterns shared between the two domains that are correlated with the text complexity:

- **Sequence length** is the most important proxy of readability;
- The use of **subordination** is highly relevant;
- The presence of **low-frequency words** impacts the sentence readability heavily.

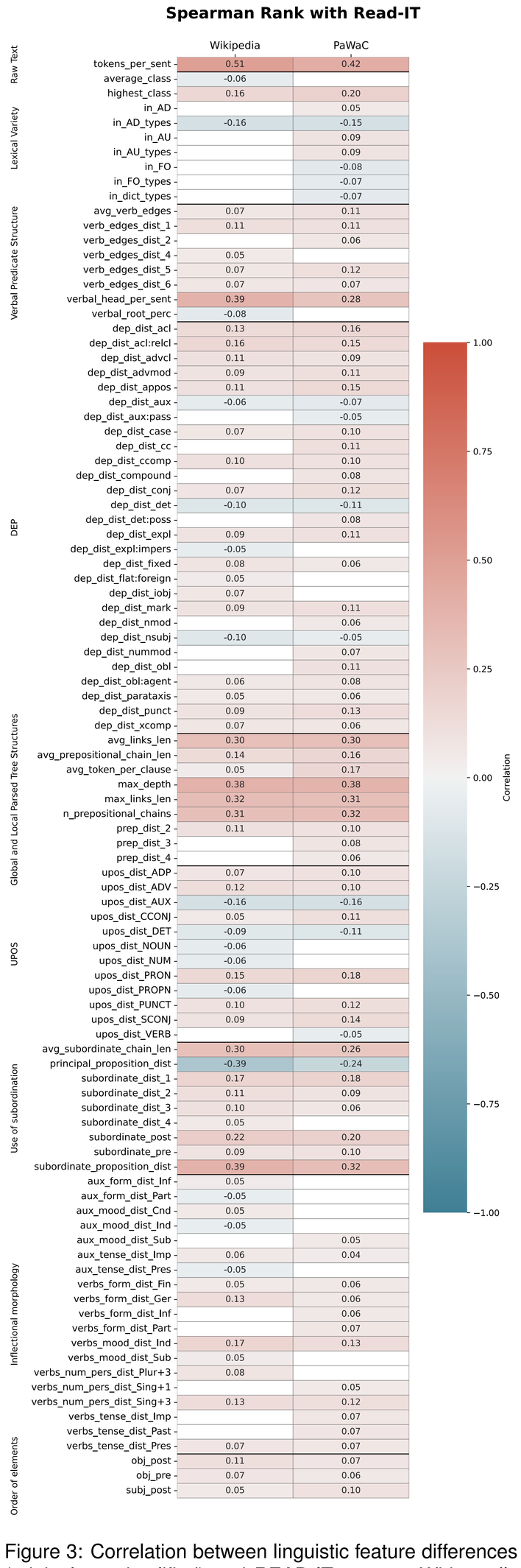


Figure 3: Correlation between linguistic feature differences (original vs. simplified) and READ-IT scores. White cells indicate non-statistically significant Spearman rank correlations (p-value < 0.05).