The Bitter Lesson Strikes Again:
Is Compute All We Need for Lexical Simplification?

Dennis Aumiller and **Michael Gertz**
Institute of Computer Science
Heidelberg University
{aumiller, gertz}@informatik.uni-heidelberg.de

Abstract

Previous state-of-the-art for lexical simplification consist of complex pipelines with several components, each of which requires deep technical knowledge to work properly. In this technical report, we describe a frustratingly simple pipeline based on prompted GPT-3 responses, beating competing approaches by a wide margin in settings with few training instances. Our best-performing submission to the English language track of the 2022 TSAR shared task consists of an "ensemble" of six different prompt templates with varying context levels. Aside from detailing the implementation and hyperparameters, we spend the remainder of this work discussing the particularities of suggestions generated by our approach and implications for future work.

1 Introduction

TODO: Cite the Bitter Lesson, the paper about emergent behavior in xLLM, and the hardware lottery wrt setup complexity/cost. TODO: Also mention details about the shared task and cite their work, detailing the current SotA a bit more. I'm pretty sure that it doesn't make sense to have a separate section for Related work on four pages, especially since I'm not super familiar with it.

2 Prompt-based Lexical Simplification

TODO: Write about what the general idea is for this model.

For the exact prompts used in our submission, please refer to Appendix A. We also detail any further hyperparameters and filtering steps used in the pipeline.

2.1 Run 1: Zero-shot Prediction

TODO: Here, we explain the basic setup of the prompting approach, and detail basic hyperparameters (ten responses per prompt that we ask the system to report). TODO: Also explain why we use the particular zero-shot prompting approach. Primarily also say that we are limited in the compute budget and the evaluation strategies on trial data with few-shot approaches. We estimate that this serves as a reasonable "lower-bound" submission. This is especially the case if we only consider pure zero-context approaches for some of the models

2.2 Filtering Predictions

TODO: Write about how we need to perform basic post-filtering because the output is not always the same. Give maybe an example in a figure? The full list of filtering operations is detailed in Appendix C.

2.3 Run 2: Ensemble Predictions

TODO: Write how we noticed that for inspections with some of the trial samples we noticed some predictions looked a bit inconsistent (or were empty). This was mostly due to the multi-word expression synonyms, which are not really what annotators provided in the survey answers

2.3.1 Utilized Prompt Setups

TODO: Basically list the six different prompt settings, and what we hope to get from this.

2.3.2 Combining Predictions

TODO: Extending this, we can obtain some of the predictions by combining the ranks across the different ensemble models

3 Error Analysis and Limitations

As with other sequence-to-sequence tasks, the output of a xLLM cannot be guaranteed to be entirely correct at all times. In this section, we detail some of the particular challenges we have encountered during the design process.

3.1 Computational Budgeting

Running a xLLM in practice, even for inference-only settings, is non-trivial and requires compute

Command	Output	Command	Output
{\"a}	ä	{\c c}	ç
{\^e}	ê	{\u g}	ğ
{\`i}	ì	{\l}	ł
{\.I}	İ	{\~n}	ñ
{\o}	ø	{\H o}	ő
{\'u}	ú	{\v r}	ř
{\aa}	å	{\ss}	ß

Table 1: Results on the English language test set of the TSAR shared task. Listed are our own results (*UniHD*), the two best-performing competing systems TODO: include, as well as provided baselines TODO: name them and cite the relevant paper.

that is far beyond many public institution's hardware budget. For the largest models with publicly available checkpoints[1], a total of around 320GB **GPU memory** is required.
The common alternative is to obtain predictions through a (generally paid) API, as was the case in this work. Especially for the ensemble model, which issues six individual requests to the API per sample, this can further bloat the net cost of a single prediction. To give context of the total cost, we incurred a total charge of $7.15 for computing predictions across the entire test set of 373 English samples for the shared task, which comes to about 1000 tokens per sample, or around $0.02 at the current OpenAI pricing scheme.[2]

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

A Prompt Templates

Table 2 provides the exact prompt templates used in the submission. Notably, the *zero-shot with context* prompt is included twice, but with different generation temperatures TODO: mention which ones, or link to the hyperparameter table?.

B Hyperparameters

We use the OpenAI Python package[3], version 0.23.0 for our experiments. For generation, the function `openai.Completion.create()` is used, where most hyperparameters remain fixed across all prompts. We explicitly list those hyperparameters below that differ from the default values. TODO: Pretty sure that most of these also constitute the default parameters? Only include those that activelsy differ!!

1. `model="text-davinci-002"`, the latest and biggest model for text generation at the time of writing,

2. `max_tokens=256`, to limit generation length,

3. `top_p=1.0`, to include the entire vocabulary during token generation,

4. `best_of=1`, TODO: I think default?,

5. `frequency_penalty=0.5`, TODO: read up on this parameter again? How is it different from the other one?,

6. `presence_penalty=0.3`, which penalizes already present tokens. We choose a lower value, since individual subword tokens might

[1] At the time of writing, this would be the 176B Bloom model TODO: cite, which has a similar parameter count to OpenAI's davinci-002 model.

[2] `https://openai.com/api/pricing/`, last accessed: 2022-10-01

[3] TODO: Insert package URL

Prompt Type	Template
Zero-shot with context	Context: {context_sentence}\n Question: Given the above context, list ten alternatives for “{complex_word}” that are easier to understand.\n Answer:
Single-shot with context	Context: A local witness said a separate group of attackers disguised in burqas — the head-to-toe robes worn by conservative Afghan women — then tried to storm the compound.\n Question: Given the above context, list ten alternative words for “disguised” that are easier to understand.\n Answer:\n1. concealed\n2. dressed\n3. hidden\n4. camouflaged\n 5. changed\n6. covered\n7. masked\n8. unrecognizable\n9. converted\n 10. impersonated\n\n Context: {context_sentence}\n Question: Given the above context, list ten alternatives for “{complex_word}” that are easier to understand.\n Answer:
Two-shot with context	Context: That prompted the military to deploy its largest warship, the BRP Gregorio del Pilar, which was recently acquired from the United States.\n Question: Given the above context, list ten alternative words for “deploy” that are easier to understand.\n Answer:\n1. send\n2. post\n3. use\n4. position\n5. send out\n 6. employ\n7. extend\n8. launch\n9. let loose\n10. organize\n\n Context: The daily death toll in Syria has declined as the number of observers has risen, but few experts expect the U.N. plan to succeed in its entirety.\n Question: Given the above context, list ten alternative words for “observers” that are easier to understand.\n Answer:\n1. watchers\n2. spectators\n3. audience\n4. viewers\n 5. witnesses\n6. patrons\n7. followers\n8. detectives\n9. reporters\n 10. onlookers\n\n Context: {context_sentence}\n Question: Given the above context, list ten alternatives for “{complex_word}” that are easier to understand.\n Answer:
Zero-shot w/o context	Give me ten simplified synonyms for the following word: {complex_word}
Single-shot w/o context	Question: Find ten easier words for “compulsory”.\n Answer:\n1. mandatory\n2. required\n3. essential\n4. forced\n 5. important\n6. necessary\n7. obligatory\n8. unavoidable\n 9. binding\n10. prescribed\n\n Question: Find ten easier words for “{complex_word}”.\n Answer:

Table 2: The exact prompt templates used for querying the model. Only \n indicate newlines, visible newlines are only inserted for better legibility. The top-most prompt template was used for Run 1, as well as part of the ensemble in Run 2. The remaining prompts were only included in the ensemble.

indeed be present several times across multiple (valid) predictions TODO: maybe explain with an example?.

TODO: Include table with the associated temperatures.

C Post-Filtering Operations

TODO: Include the list of operations by which we filter.