UniHD at TSAR-2022 Shared Task: 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 models for lexical simplification consist of complex pipelines with several components, each of which requires deep technical knowledge and fine-tuned interaction to achieve its full potential. As an alternative, 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 TSAR-2022 shared task consists of an "ensemble" of six different prompt templates with varying context levels. As a late-breaking result, we further detail a language transfer technique that allows simplification in languages other than English. Applied to the Spanish and Portuguese subset, we achieve state-of-the-art results with only minor modification to the original prompts. Aside from detailing the implementation and setup, we spend the remainder of this work discussing the particularities of prompting and implications for future work.

1 Introduction

With recent advancements in Machine Learning (ML) research coming largely from increasing compute budgets, Richard Sutton coined the idea of a "bitter lesson", wherein more computational power will ultimately supersede a hand-crafted solution (Sutton, 2019). More recently, increasing compute on a general purpose architecture has also shown to be wildly successful in the Natural Language Processing (NLP) community (Vaswani et al., 2017; Wei et al., 2022). In particular, emergent capabilities in very large language models (vLLMs) have made it possible to approach a variety of tasks wherein only few (if any) samples are provided with labels, and no further fine-tuning on task-specific data is required at all.

In stark contrast to the complex setup required for modern lexical simplification systems (Ferrés et al., 2017; Qiang et al., 2020; Štajner et al., 2022), we present a simplistic pipeline based on simply prompting a vLLM with basic instructions on simplification, which returns frustratingly good results considering the overall complexity of the approach. Results at the TSAR-2022 shared task (Saggion et al., 2022) rank our model as the best-performing one on the English subset, indicate that this approach may be another instance of the bitter lesson. While the initial findings are indeed promising, we want to cautiously evaluate erroneous instances on the test set to analyze potential pitfalls, and further detail some of our experiences in crafting prompts for context-dependent prediction tasks. Furthermore, we acknowledge the technical challenges in reproducing (and productionizing) systems based on vLLMs, especially given that suitable models exceed traditional computing budgets.

TODO: somewhere I need to actually mention more details on the TSAR shared task and task data!! This could be enough content for a Background/Rel Work section?

2 Prompt-based Lexical Simplification

TODO: Write about what the general idea is for this model. Touted as the TODO: insert flashy phrase, such as "human-level AI", the GPT-3 model released by OpenAI (Brown et al., 2020) has been the first in a series of available vLLMs for general-purpose text generation (Thoppilan et al., 2022; BigScience, 2022; Zhang et al., 2022). Across these models, a general trend in scaling beyond a particular parameter size can be observed, while keeping the underlying architectural design close to existing smaller models TODO: maybe cite something. Through exhibiting zero-shot transfer capabilities, such models have also become more attractive for lower-resourced tasks. Often, models are able to answer questions formulated in natural language with somewhat sensible results. Particular template patterns (so-called *prompts*) are fre-

quently used to guide models towards predicting a particularly desirable output or answer format.

Using this paradigm, we experimented with different prompts issued to OpenAI's largest available model, `text-davinici-002`, which totals 176B parameters. Our first approach uses a singular prompt template in a zero-shot setting to obtain predictions for the shared task; we further improve upon these results by combining predictions from different prompt templates later on.

2.1 Run 1: Zero-shot Prediction

As a baseline system with a performance indication that is realistic for 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

Generated suggestions generally are mostly returned as a comma-separated list or enumeration with line breaks. However, in our experience a clear pattern cannot be expected from the model and seems to be non-deterministic even with set template structures. We parse available outputs in the known formats and re-run samples with differing structures, which seems to resolve the issue in our case. Additionally, we employ a list of simple filters to ensure the quality of predictions, as detailed in Appendix C.

2.3 Run 2: Ensemble Predictions

Upon inspecting the results from the first run, we noticed that in some instances, predictions were almost fully discarded due to filtering. Simultaneously, we had already previously encountered strong variance in system generations when varying the prompt template or altering the context setting. To this extent, an ensemble of predictions from multiple different prompt templates was utilized to broaden the spectrum of possible generations, as well as ensuring that a minimum number of suggestions survives the filtering step.

2.3.1 Prompt Variants

The exact prompts are detailed in Table 2. Utilized template variants can be grouped into *with context* (the context sentence is provided), or *without context* (synonyms are generated from the complex word alone). Simultaneously, different prompts also contain between zero and two examples taken from the trial data, including their expected outputs. This can be interpreted as a few-shot setting in which the model is demonstrated on what correct answers may look like for this particular task. Aside from this, we vary the generation temperature, where a higher temperature increases the likelihood of a potentially more creative (but not always correct) prediction.

2.3.2 Combining Predictions

For each of the six prompts p, we ask the model to suggest ten alternative simplified expressions S_p and filter them with the exact same rules as the single prompt system in Run 1. In order to combine and re-rank suggestions s, we assign a combination score V to each distinct prediction $s \in \bigcup_p S_p$:

$$V(s) = \sum_p \max(5.5 - 0.5 \cdot \text{rank}_{S_p}(s), 0), \quad (1)$$

where $\text{rank}_{S_p}(s)$ is the ranked position of suggestion s in the resulting ranking from prompt p. If $s \notin S_p$, we set $\text{rank}_{S_p}(s) = \infty$. The scaling parameters are chosen arbitrarily and can be adjusted to account for the expected number of suggestions per prompt.

3 Error Analysis and Limitations

As with other sequence-to-sequence tasks, the output of a vLLM 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 vLLM in practice, even for inference-only settings, is non-trivial and requires compute that is far beyond many public institution's hardware budget. For the largest models with publicly available checkpoints[1], a total of around **325GB GPU memory** is required, assuming efficient storage in `bfloat16` or similar precision levels.

[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.

Run	ACC@1	Acc@k@Top1			MAP@k			Potential@k		
		$k=1$	$k=2$	$k=3$	$k=3$	$k=5$	$k=10$	$k=3$	$k=5$	$k=10$
Ensemble (Ours)	**0.8096**	**0.4289**	**0.6112**	**0.6863**	**0.5834**	**0.4491**	**0.2812**	**0.9624**	**0.9812**	**0.9946**
Single (Ours)	0.7721	0.4262	0.5335	0.5710	0.5090	0.3653	0.2092	0.8900	0.9302	0.9436
MANTIS-1	0.6568	0.319	0.4504	0.5388	0.473	0.3599	0.2193	0.8766	0.9463	0.9785
UoM&MMU-1	0.6353	0.2895	0.4530	0.5308	0.4244	0.3173	0.1951	0.8739	0.9115	0.9490
LSBert	0.5978	0.3029	0.4450	0.5308	0.4079	0.2957	0.1755	0.8230	0.8766	0.9463
TUNER	0.3404	0.1420	0.1689	0.1823	0.1706	0.1087	0.0546	0.4343	0.4450	0.4450

Table 1: Results on the English language test set of the TSAR-2022 shared task, ranked by *ACC@1* scores. Listed are our own results (*Ensemble* and *Single*), the two best-performing competing systems (*MANTIS* and *UoM&MMU*), as well as provided baselines (*LSBert* (Qiang et al., 2020) and TUNER (Ferrés et al., 2017)).

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 slightly over $7 for computing predictions across the entire test set of 373 English samples, which comes out to about 1000 tokens per sample, or around $0.02 at the current OpenAI pricing scheme.[2] For the Spanish subset and language-dependent prompt development, the total cost came to about $10, primarily due to longer sample contexts. Costs for Portuguese processing were around $6.50. Aside from monetary concerns, environmental impacts are also to be considered for larger-scale deployments of this kind TODO: cite the environmental paper everyone brings up.

4 Conclusion and Future Work

TODO: Maybe ask Matthew/organizers if we can actually get the predictions for the baselines? This way, we could compare the approaches. The context paper also mentions a few instances, so maybe look there? Utilizing prompted responses from vLLMs seems to be

References

Workshop BigScience. 2022. BLOOM (revision 4ab0472).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Ferrés, Horacio Saggion, and Xavier Gómez Guinovart. 2017. An adaptable lexical simplification architecture for major Ibero-Romance languages. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 40–47, Copenhagen, Denmark. Association for Computational Linguistics.

Jipeng Qiang, Yun Li, Zhu Yi, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. *Thirty-Fourth AAAI Conference on Artificial Intelligence*, page 8649–8656.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the tsar-2022 shared task on multilingual lexical simplification. In *Proceedings of TSAR workshop held in conjunction with EMNLP 2022*.

Richard Sutton. 2019. The bitter lesson. *Incomplete Ideas (blog)*, 13:12.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton,

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

Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Sanja Štajner, Daniel Ferrés, Matthew Shardlow, Kai North, Marcos Zampieri, and Horacio Saggion. 2022. Lexical simplification benchmarks for English, Portuguese, and Spanish. *Frontiers in Artificial Intelligence*, 5.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *CoRR*, abs/2206.07682.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

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?. For few-shot prompts, we have taken samples from the previously published trial set for the respective language. In instances where less than 10 distinctly different suggestions were provided by annotators, we manually extended the list of examples to match exactly ten results based on our own judgment. For instances with more provided suggestions, we limit ourselves to the ten most frequently occurring ones. The reason for this is that GPT-3 otherwise tended to return an inconsistent number of suggestions in our preliminary testing.

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.

1. `model="text-davinci-002"`, which is the latest and biggest available model for text completion.

2. `max_tokens=256`, to ensure sufficient room for generated outputs. In practice, most completions are vastly below the limit.

3. `frequency_penalty=0.5`, as well as `presence_penalty=0.3`, which jointly penalize present tokens and token repetitions. The values are well below the maximum (values can range from -2 to 2), since individual subword tokens might indeed be present several times across multiple (valid) predictions. A more detailed computation can be found in the documentation of OpenAI.[4]

Outside of the repetition penalties, the most influential parameter choice for generation is the sampling temperature. We generally take a more measured approach than the default (`temperature=1.0`), but vary temperature across our ensemble prompts to ensure a more diverse result set overall. We list the used temperatures in Table 2. *Zero-shot with context* is used twice in the ensemble, once with a more conservative temperature, and once with a more "creative" (higher) temperature. For the singular prompt run, we use the conservative *zero-shot with context* variant.

C Post-Filtering Operations

Given the uncertain nature of predictions by a language model, we employ a series of post-filtering steps to ensure high quality outputs. This includes stripping newlines/spaces/punctuation (`\n :;.?!`), lower-casing, removing infinitive forms ("to deploy" instead of "deploy" as a prediction), as well as removing identity predictions (e.g., the prediction being the same as the original complex word) and deduplicating suggestions. Additionally, we noticed that for some instances, generated synonyms resemble more of a "description" rather than truly synonymous expressions (example: "people that are crazy" as a suggestion for "maniacs"). Given the nature of provided data, we removed

[3] TODO: Insert package URL

[4] https://beta.openai.com/docs/api-reference/parameter-details

Prompt Type	Template

Zero-shot with context
temperature:
0.3 (conservative),
0.8 (creative)
	`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
temperature: 0.5
	`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\n5. changed\n6. covered\n7. masked\n8. unrecognizable\n9. converted\n10. 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
temperature: 0.5
	`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\n6. 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\n5. witnesses\n6. patrons\n7. followers\n8. detectives\n9. reporters\n10. 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 temperature: 0.7	`Give me ten simplified synonyms for the following word: {complex_word}`

Single-shot w/o context
temperature: 0.6
	`Question: Find ten easier words for “compulsory”.\n`
	`Answer:\n1. mandatory\n2. required\n3. essential\n4. forced\n5. important\n6. necessary\n7. obligatory\n8. unavoidable\n9. binding\n10. prescribed\n\n`
	`Question: Find ten easier words for “{complex_word}”.\n`
	`Answer:`

Table 2: The English prompt templates used for querying the OpenAI model, including associated generation temperatures. Only written out “\n” symbols indicate newlines, visible line breaks are inserted for better legibility. Only top-most prompt template with conservative temperature was used in the single prompt (Run 1), as well as in the ensemble run (Run 2). All other prompts‘ were only included in the ensemble submission.

extreme multi-word expressions (for English, any suggestion with more than two words, for Spanish and Portuguese more than three words in a single expression).

D Results for Spanish and Portuguese

Given the surprisingly good results on the English subset, we decided to extend our experiments to the Spanish and Portuguese tracks as well. Given that we did not submit any original runs, we did not include them in the main body of the paper. Transferring the prompts to Spanish or Portuguese is surprisingly simple. We alter the prompt to:
*"Given the above context, list ten alternative **Spanish** words for 'complex_word' that are easier to understand."* (bold highlight indicates change)
Without this change, returned suggestions generally tend to be in English, which could be an attractive opportunity to mine cross-lingual simplifications in future work. By adding the output language explicitly, we ensure that the suggestions match the expected results. For Portuguese, the prompt can be adapted accordingly.
We find that our system also outperforms all competing submitted approaches in the shared task; result comparisons can be found in Table 3 and Table 4, respectively.

Run	ACC@1	Acc@k@Top1			MAP@k			Potential@k		
		$k=1$	$k=2$	$k=3$	$k=3$	$k=5$	$k=10$	$k=3$	$k=5$	$k=10$
Ensemble (Ours)	**0.6521**	**0.3505**	**0.5108**	**0.5788**	**0.4281**	**0.3239**	**0.1967**	**0.8206**	**0.8885**	**0.9402**
Single (Ours)	0.5706	0.3070	0.3967	0.4510	0.3526	0.2449	0.1376	0.6902	0.7146	0.7445
PresiUniv-1	0.3695	0.2038	0.2771	0.3288	0.2145	0.1499	0.0832	0.5842	0.6467	0.7255
UoM&MMU-3	0.3668	0.1603	0.2282	0.269	0.2128	0.1506	0.0899	0.5326	0.6005	0.6929
LSBert	0.2880	0.0951	0.1440	0.1820	0.1868	0.1346	0.0795	0.4945	0.6114	0.7472
TUNER	0.1195	0.0625	0.0788	0.0842	0.0575	0.0356	0.0184	0.144	0.1467	0.1494

Table 3: Results on the Spanish language test set of the TSAR-2022 shared task, ranked by *ACC@1* scores. Listed are our own results (*Ensemble* and *Single*), the two best-performing competing systems (*PresiUniv* and *UoM&MMU*), as well as provided baselines (*LSBert* (Qiang et al., 2020) and TUNER (Ferrés et al., 2017)).

Run	ACC@1	Acc@k@Top1			MAP@k			Potential@k		
		$k=1$	$k=2$	$k=3$	$k=3$	$k=5$	$k=10$	$k=3$	$k=5$	$k=10$
Ensemble (Ours)	**0.7700**	**0.4358**	**0.5347**	**0.6229**	**0.5014**	**0.3620**	**0.2167**	**0.9171**	**0.9491**	**0.9786**
Single (Ours)	0.6363	0.3716	0.4625	0.5160	0.4105	0.2889	0.1615	0.7860	0.8181	0.8422
GMU-WLV-1	0.4812	0.2540	0.3716	0.3957	0.2816	0.1966	0.1153	0.6871	0.7566	0.8395
Cental-1	0.3689	0.1737	0.2433	0.2673	0.1983	0.1344	0.0766	0.524	0.5641	0.6096
LSBert	0.3262	0.1577	0.2326	0.286	0.1904	0.1313	0.0775	0.4946	0.5802	0.6737
TUNER	0.2219	0.1336	0.1604	0.1604	0.1005	0.0623	0.0311	0.2673	0.2673	0.2673

Table 4: Results on the Portuguese language test set of the TSAR-2022 shared task, ranked by *ACC@1* scores. Listed are our own results (*Ensemble* and *Single*), the two best-performing competing systems (*MANTIS* and *UoM&MMU*), as well as provided baselines (*LSBert* (Qiang et al., 2020) and TUNER (Ferrés et al., 2017)).