# RIGA at SemEval-2023 Task 2: NER enhanced with GPT-3

**Eduards Mukans**
mukans.work@gmail.com
University of Latvia, Faculty of Computing

**Guntis Barzdins**
guntis.barzdins@lumii.lv
University of Latvia, IMCS

## Abstract

The following is a description of the RIGA team's submissions for the English track of the SemEval-2023 Task 2: Multilingual Complex Named Entity Recognition (MultiCoNER) II. Our approach achieves 17% boost in results by utilizing pre-existing Large-scale Language Models (LLMs), such as GPT-3, to gather additional contexts. We then fine-tune a pre-trained neural network utilizing these contexts. The final step of our approach involves meticulous model and compute resource scaling, which results in improved performance. Our results placed us 12th out of 34 teams in terms of overall ranking and 7th in terms of the noisy subset ranking. The code for our method is available on GitHub[1].

## 1 Introduction

The SemEval 2023 Task 2: MultiCoNER II (Multilingual Complex Named Entity Recognition) (Fetahu et al., 2023b) presented participants with a challenge to classify words into 36 named and nominal entities. Unlike traditional named entity recognition (NER) tasks that deal with relatively simple classes such as persons, locations, and organizations, the MultiCoNER task involves entities with any linguistic constituents. These include entries such as creative works, food, and medical procedures, among others. Furthermore, rather than simply classifying a word as part of a high-level group, the task requires participants to choose the specific type within the group to which the word belongs. For instance, if the word pertains to a person, it should be classified as an artist, an athlete, a politician, and so on.

The datasets provided for the MultiCoNER II Task included 12 languages, with participants having the flexibility to submit their results for any language of their choosing. In our submission, we focused solely on the most competitive English track and did not submit to other languages.

The task at hand is considerably more complex when compared to traditional NER tasks, as all input texts are lower-cased. Thus, our submission primarily relied on the utilization of pre-existing pre-trained large language models (LLMs).

The history of machine learning compute can be divided into three eras, namely the Pre-Deep Learning Era, the Deep Learning Era, and the Large-Scale Era (Sevilla et al., 2022). The Large-Scale Era was initiated by the release of AlphaGo (Silver et al., 2016) in late 2015. This trend is likely to have emerged due to large corporations identifying new opportunities in the field and making significant investments in it, thereby resulting in considerable increases in model training budgets. This, in turn, marked the beginning of a new era of large-scale language models. Concurrently, it also prompted small institutes with limited budgets to explore novel methods for parameter efficient fine-tuning (PEFT) of LLMs developed by large corporations.

Several approaches have been proposed to address the issue of PEFT for LLMs. One such method involves incorporating small bottleneck layers referred to as adapters (Bapna and Firat, 2019; Houlsby et al., 2019; Pfeiffer et al., 2021) into the original LLM. In this technique, all parameters of the original LLM are frozen, and only the bottleneck layers are trainable. The approach involves adding just 3-5% of trainable parameters on top of the model, resulting in outcomes comparable to full fine-tuning. While it is true that full fine-tuning may yield marginally better results than adapters, we switched to full fine-tuning only at later steps.

The task of selecting a specific entity type out of a group is a challenging one, even for humans, as it requires a considerable amount of background knowledge on various topics. Without additional context about the sentence, correctly identifying

[1] https://github.com/emukans/multiconer2-riga

*Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 331–339
July 13-14, 2023 ©2023 Association for Computational Linguistics

the entity type by merely looking at a few words is nearly impossible. Thus, in our submission, we focused on leveraging the GPT-3 model (Brown et al., 2020) to enhance token class prediction.

The authors of (Brown et al., 2020) proposed a method called "in-context learning", which involves presenting the model with a task template with a blank gap that it must fill in. In our case, we utilized the GPT-3 language model to mine additional context about the sentences. As the GPT-3 language model is trained on data up to June 2021 (OpenAI, 2023), its general knowledge is up to date, and it is aware of the most significant events up to that time.

## 2 Related Work

During SemEval-2022 (Malmasi et al., 2022), the top-performing systems employed external knowledge bases (Wang et al., 2022; Ma et al., 2022) and gazetteers (Chen et al., 2022) to tackle the challenge. However, these solutions are prone to inaccuracies when faced with entities outside of the knowledge base and in scenarios where there are spelling errors and typos. In SemEval-2023 (Fetahu et al., 2023b), the authors highlight the limitations of the previous year's best-performing models in addressing these issues.

Recent advancements in natural language processing (NLP) have led to the development of transformer-based language models (LMs) such as BERT (Devlin et al., 2019) and, its larger and more advanced multi-lingual variant, XLM-R (Conneau et al., 2019). These models have achieved further improvements in NLP tasks by utilizing deep contextual word representations.

The difficulties of NER in real-world scenarios have been addressed in recent studies such as (Meng et al., 2021) and (Fetahu et al., 2021). These studies employ external gazetteer knowledge bases to identify entities in low-context environments.

Typically, modern neural networks are trained using early stopping techniques (Prechelt, 2012), where the training process is halted when the loss function starts to increase on unseen validation data. In the past year, our team participated in the CODWOE task of SemEval (Mickus et al., 2022), where we explored various training effects such as "Deep Double Descent" (Nakkiran et al., 2019) and model "Scaling Laws" (Kaplan et al., 2020). The "Scaling Law" effect occurs when two out of three training aspects, namely data, parameter count, and compute, are scaled. Since the data amount was fixed, we scaled only the model size and compute time, and achieved an epoch-wise deep double descent. Furthermore, we found that continuing training past the overfitting point can lead to more robust model predictions. In last year's competition, we submitted to several languages and achieved 1st place for French, 2nd place for Spanish, and 3rd place for Russian. These results demonstrate that loss is not always correlated with accuracy or F1 scores, and that stopping training too early can hurt performance.

This year, we replicated our approach and scaled the number of trainable parameters and the amount of compute time. The results of training without early stopping have shown a significant improvement compared to the results obtained using early stopping.

## 3 Data

In comparison to the previous iteration of the task, the new challenge in the latest dataset (Fetahu et al., 2023a) is the presence of noisy sentences. Named entities in the dataset may contain typographical errors, missing named entity labels, or inaccurate labels. For instance, instead of being labeled as an `Athlete`, the entity may be labeled as `OtherPER`, indicating that the entity is an unknown type of person.

The `multiconer2-data` dataset was generated semi-automatically, which resulted in missing labels for some named entities. Specifically, in some sentences that contained two named entities, the training and development datasets lacked labels for one or more of them.

Upon analysis of the `multiconer2-data` dataset, it was discovered that some of the sentences were duplicated. It is unclear whether the dataset creators intended to introduce bias into the model training or if the duplicates were accidental. The train dataset contains 0.7% duplicated data, while the test dataset contains 1.7% duplicated data.

The complexity of the task is further compounded by the unbalanced frequency of named entities, as detailed in the distribution shown in the appendix A, and the size of the datasets, as indicated in Table 1.

As the size of the `Dev` dataset is small, it is difficult to determine the significance of the difference in experimental results. Therefore, another dataset,

| Dataset size | Train | Validate | Dev | Test |
|---|---|---|---|---|
| Original | 16.8K | 0 | 0.9K | 250K |
| Cleaned | 13.3K | 3.3K | 0.9K | 250K |

Table 1: Dataset size distribution

called `Validation`, was created from the cleaned `Train` dataset. During the cleaning process, duplicated sentences were reduced to a single unique entry.

# 4 Methodology

## 4.1 Data preprocessing

In the NER task, our approach involved three steps. Initially, we performed data cleaning and preprocessing. The original `multiconer2-data` dataset was in CoNLL format, and we converted it to HuggingFace Datasets (Lhoest et al., 2021) for easier use.

## 4.2 Gathering context

In light of previous winning submissions utilizing external knowledge bases, we opted to gather additional context for our approach this year. However, instead of developing a custom knowledge base and a separate microservice on ElasticSearch like the previous winning system (Wang et al., 2022), we chose a more elegant approach. We outsourced the knowledge base and search mechanism to the GPT-3 network (Brown et al., 2020). Hence, in the second step, we extracted additional contexts for every sentence in the dataset.

To obtain additional context, we employed the "in-context learning" method, which was introduced in the original GPT-3 research paper (Brown et al., 2020). This involves designing an input prompt template and tuning the model parameters to encourage the generative neural network to produce the desired output. This process is referred to as "prompt engineering".

At the time of competition and writing this paper, we utilized the largest GPT-3 model available, which is `text-davinci-003`, in order to achieve the best results in our experiments and submissions.

Before commencing with the tuning of model parameters, it is imperative to make our intentions clear. Since our objective is to mine extra context, it is necessary that the output is precise and robust to a high degree. Therefore, we need to limit the degree of experimentation that we can perform on the model.

The OpenAI model also includes a set of tunable parameters, including temperature, max length, top P, frequency penalty, presence penalty, and best of.

The "Best of" parameter determines how many completions are generated by the server, but only the best one is displayed. To conserve token usage, we kept this parameter constant at 1.

The "Top P" parameter controls diversity through the use of nucleus sampling (Holtzman et al., 2020). If the parameter value is less than 1, the model considers all samples where the probability is greater than the "Top P" value. To ensure consistent and accurate output, we maintained the "Top P" parameter at a fixed value of 1.

The "Presence penalty" parameter imposes a penalty on new tokens that appear in the generated text. Since named entities are more likely to be repeated, and the repetition can help the model identify them, we want to encourage token repetition. Therefore, we fixed the parameter and set it to 0.

The "Temperature" parameter controls the level of randomness in the generated text. A lower temperature forces the model to produce more deterministic and repetitive results. However, for our task of using the neural network as a knowledge base, the results need to be deterministic. Therefore, we fixed the temperature parameter to 0.2 to make the outputs more predictable while still leaving some room for creativity and randomness.

The "max length" parameter determines the maximum number of tokens that the model can generate in the output. The "frequency penalty" parameter regulates how much to penalize new tokens based on their frequency in the text generated so far. Both of these parameters were adjustable, and we have summarized some of the results of our experiments in appendix B. Although we had more samples during our experiments, we included only representative examples in the table. In other attempts, either the outputs were very similar or there was no discernible difference.

By changing the "frequency penalty" parameter from 0 to 0.5, we observed that the named entities had the same frequency. This result could be related to the fact that the "presence penalty" was set to 0. When we increased the "frequency penalty," we noticed that the text became more versatile and varied. During the context mining process, we set the penalty to 0.5.

We conducted experiments with "max length" parameters of 96 and 128. From the examples, we observed that a larger "max length" parameter resulted in longer generated context, but the generated text did not provide more relevant information about the input sentence. Thus, for the later context mining process, we set the "max length" to 96. We also attempted to use smaller values, but the sentences were often truncated and the intended meaning was not fully developed.

### 4.3 NER model training

In the initial stage, we conducted numerous attempts with various training data manipulations. To optimize resource utilization and enable parallel experimentation, we employed adapter modules (Houlsby et al., 2019) on top of the pre-trained `bert-base-uncased` (BERT) model (Devlin et al., 2019). The F1 score was our primary metric for evaluation, and unless otherwise specified, we evaluated our approach on the `Validation` dataset. The `Dev` dataset, being too small to yield meaningful conclusions, was used only in our final attempts before submitting our work for evaluation. We used four Tesla v100 16GB GPUs, provided by our institution, for conducting these experiments. The results of our experiments, presented in table 2, reflect our efforts to fine-tune the pre-trained model for the NER task.

In our initial attempt, we sought to simplify the NER task by training the model in multiple steps. Firstly, we trained a simplified NER with only six classes, where each of the specific classes such as `Athlete`, `Artist`, `Scientist`, etc., were grouped under a top-level class, i.e., `Person`. We then stacked another adapter layer on top of it using the adapter fusion method (Pfeiffer et al., 2021). In this method, the pre-trained adapter and the pre-trained model were frozen. However, the results were not successful, and the best score obtained was slightly lower than the plain model training without any modifications. The baseline score for adapter training was 63.2%, and the best score achieved was also 63.2%.

Later we attempted the same approach as last year without early stopping. The resulting model, which used adapters but did not include early stopping, is labeled as `adapter-no-early`. Unfortunately, simply scaling the compute time did not produce the expected improvements in performance.

The use of adapter modules can improve the efficiency and robustness of training. The loss value has a positive correlation with the F1 score, which explains the lack of difference between the `adapter-baseline` and `adapter-no-early` experiments. On the other hand, full fine-tuning involves all parameters in the training process, and the baseline for full fine-tuning with early stopping is denoted as `full-no-context-baseline` and achieves a score of 61.2%. However, the F1 plot for this experiment indicates that it does not correlate with the loss value, and early stopping is triggered too early. Removing early stopping allows the model to train longer and achieve a much better result. The full fine-tuning without early stopping is denoted by `full-no-early` and has a score of 69.9%. It is important to note that the scaling laws only work when at least two out of three training aspects are increased. Now we could confirm, that it works when you scale the model **trainable** parameters, not the total amount of parameters.

When comparing the adapter version to the plain pre-trained model, it can be observed that the adapter version has more parameters - 109.8M, while the plain model without adapters has 108.9M parameters. The adapter version is larger due to the inclusion of bottleneck layers in the model, with only these added layers being trainable. In the adapter experiment, only 0.9M (0.8%) of parameters were trainable, whereas in full fine-tuning, all 108.9M parameters were trainable.

In the subsequent experiment iteration, we introduced additional context that was obtained through the use of the GPT-3 model API. Consequently, the input was structured as follows: `"<s>[input sentence]</s>[mined context]</s>"`. !NB, the start and end tokens as well as the separator tokens may vary depending on the selected model tokenizer. The output is a sequence, where each input token gets assigned tag or `O` which denotes no tags. The key for the OpenAI contexts is `full-openai-contexts`. Incorporating the extra context resulted in an additional 4% improvement in performance.

In a separate experiment, we attempted to predict named entities using GPT-3. We gathered all contexts and then the new token classification model is trained, which accepts input sentence and GPT-3 generated prediction. This experiment is labeled as `full-openai-entities`, and the results were worse than those without any additional information. We hypothesize that this is due to the genera-

| Key | Pre-trained model | Total params, M | Trainable params, % | F1, % |
|---|---|---|---|---|
| adapter-baseline | bert-base-uncased | 109.8 | 0.8 | 63.4 |
| adapter-fusion | | 140.9 | 22.7 | 63.2 |
| adapter-no-early | | 109.8 | 0.8 | 63.4 |
| full-no-context-baseline | | 108.9 | 100 | 61.2 |
| full-no-early | | 108.9 | 100 | 69.9 |
| full-openai-contexts | | 108.9 | 100 | **74.0** |
| full-openai-entities | | 108.9 | 100 | 67.6 |
| full-crf-head | | 108.9 | 100 | 72.4 |
| adapter-baseline-large | xlm-roberta-large | 584.1 | 4.3 | 64.0 |
| full-baseline-large | | 558.9 | 100 | 67.4 |
| full-tuned-no-early | | 558.9 | 100 | **79.7** |

Table 2: Training NER neural network. The scores measured on `Validation` dataset

| Type | Clean Subset F1, % | Noisy Subset F1, % | Overall Macro F1, % |
|---|---|---|---|
| Official Baseline | N/A | N/A | 36.97 |
| No contexts | N/A | N/A | 52.36 |
| With contexts | 70.74 | 66.07 | **69.3** |

Table 3: Submission results

tive model's inherent noisiness, which may produce numerous incorrect answers.

In NER, it is a common practice to use Conditional Random Fields (CRFs) (Lafferty et al., 2001) instead of simple multi-layer perceptrons (MLPs). Therefore, we also experimented with CRFs by using them as the head of our model. However, this experiment, which we labeled as `full-crf-head`, did not provide any significant improvements.

Finally, according to the scaling laws, we increased the model size from `bert-base-uncased` (BERT) to `xlm-roberta-large` (XLM-R) and extended the training time in the next experiment. The key for this experiment is `full-tuned-no-early`. Additionally, we paid close attention to tuning the hyperparameters, which is why the word `tuned` is included in the key. The results for this model were the best, with a score of 79.7%, and we subsequently used this model to submit our final rankings.

## 5 Results

We submitted two entries for the final ranking, one with additional contexts and the other without. The results of both entries are presented in Table 3. Our submission was ranked 12th among 34 teams that participated in the English track.

Regrettably, the organizers calculated the F1 scores for the Clean and Noisy subsets only for the top-performing solution. Therefore, we have no information about these scores for the submissions without contexts.

When comparing the submissions with and without contexts, a significant boost in performance is observed, with almost a 17-point increase. Furthermore, it is concluded that GPT-3 is efficient in handling noise, leading to a small difference between the clean and noisy subsets. Ranking based on the noisy subset elevates our score to the 7th position.

We evaluated alternative language models, such as LLaMA (Touvron et al., 2023), which became available subsequent to the conclusion of the competition. However, the resultant contexts produced by LLaMA exhibited comparatively lower levels of semantic coherence when contrasted with outputs derived from GPT-3.

## 6 Conclusion

Throughout the competition, our team confirmed the hypothesis that additional context significantly impacts the accuracy of named entity recognition. This highlights the importance for researchers to leverage existing Large-scale Language Models, such as GPT-3, when designing solutions as it can greatly enhance overall performance and stability in a noisy environment.

Utilizing the scaling laws is one of the simplest ways to boost system performance. By scaling at least two out of three aspects of model training, the system outcome can be improved.

## Acknowledgements

The research was supported by the EU Recovery and Resilience Facility project Language technology Initiative (No 2.3.1.1.i.0/1/22/I/CFLA/002), by the H2020 project SELMA (under grant agreement No. 957017), and by the Latvian Council of Science project lzp-2021/1-0479.

## References

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

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.

Beiduo Chen, Jun-Yu Ma, Jiajun Qi, Wu Guo, Zhen-Hua Ling, and Quan Liu. 2022. USTC-NELSLIP at SemEval-2022 Task 11: Gazetteer-Adapted Integration Network for Multilingual Complex Named Entity Recognition. *arXiv e-prints*, page arXiv:2203.03216.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. MultiCoNER v2: a Large Multilingual dataset for Fine-grained and Noisy Named Entity Recognition.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. Gazetteer Enhanced Named Entity Recognition for Code-Mixed Web Queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681.

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023b. SemEval-2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Long Ma, Xiaorong Jian, and Xuan Li. 2022. PAI at SemEval-2022 task 11: Name entity recognition with contextualized entity representations and robust loss functions. In *Proceedings of the 16th International*

*Workshop on Semantic Evaluation (SemEval-2022)*, pages 1665–1670, Seattle, United States. Association for Computational Linguistics.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2019. Deep double descent: Where bigger models and more data hurt. *CoRR*, abs/1912.02292.

OpenAI. 2023. The openai model api.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Lutz Prechelt. 2012. *Early Stopping — But When?*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. 2022. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022. DAMO-NLP at SemEval-2022 task 11: A knowledge-based system for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1457–1468, Seattle, United States. Association for Computational Linguistics.

## A Tag distribution in train dataset

| Tag | % | Tag | % | Tag | % |
|---|---|---|---|---|---|
| Artist | 15 | PublicCORP | 2 | SportsManager | 1 |
| HumanSettlement | 10 | AnatomicalStructure | 2 | Disease | 1 |
| OtherPER | 7 | Software | 2 | Vehicle | 1 |
| Athlete | 7 | Station | 2 | CarManufacturer | 1 |
| ORG | 6 | Food | 1 | Scientist | 1 |
| VisualWork | 5 | OtherLOC | 1 | Symptom | 1 |
| MusicalWork | 4 | Medication/Vaccine | 1 | AerospaceManufacturer | 1 |
| WrittenWork | 4 | Clothing | 1 | | |
| Facility | 4 | MedicalProcedure | 1 | | |
| OtherPROD | 3 | Drink | 1 | | |
| MusicalGRP | 3 | Cleric | 1 | | |
| SportsGRP | 3 | ArtWork | 1 | | |

## B Gathered OpenAI contexts

| Input sentence | Max length | Frequency penalty | Context |
|---|---|---|---|
| the species was described by dietrich brandis after the forester t. f. bourdillon. | 128 | 0 | Named entities in this text are "species", "Dietrich Brandis", and "T.F. Bourdillon". Dietrich Brandis was a German-born British botanist and forester who was the first Inspector General of Forests in British India. T.F. Bourdillon was a British forester who worked in India and is credited with introducing the concept of scientific forestry to the country. The species mentioned in the text is likely a species of plant or tree that was discovered by Bourdillon and described by Brandis. |
| | | 0.3 | Named entities in this text are "species", "Dietrich Brandis", and "T. F. Bourdillon". Dietrich Brandis was a German forester and botanist who worked in India and is credited with introducing the science of forestry to the country. T. F. Bourdillon was a British forester who worked in India and is known for his work on the teak forests of Burma. The species mentioned was likely a species of plant or animal that was discovered by Bourdillon and described by Brandis. |
| | | 0.5 | Named entities in this text are "species," "Dietrich Brandis," and "T.F. Bourdillon." Dietrich Brandis was a German-born British botanist and forester who is credited with introducing the concept of scientific forestry to India and Southeast Asia. T.F. Bourdillon was an Indian forester who worked with Brandis in India and is credited with helping to establish the first forest school in India. The species mentioned in the text was likely a plant or animal species discovered by Brandis and/or Bourdillon during their work together in India. |
| boris sidis groundbreaking work on the psychology of suggestion dissociative identity disorder psychopathology and genius | 96 | 0.5 | Boris Sidis was a pioneering psychologist and neurologist who made significant contributions to the fields of psychology and psychiatry. He is best known for his groundbreaking work on the psychology of suggestion, dissociative identity disorder, psychopathology, and genius. His research focused on the study of abnormal behavior, mental illness, and the development of genius. He was also an early advocate for psychotherapy as a treatment for mental illness. His work has had a lasting impact on modern psychology |
| | 128 | | Boris Sidis was a pioneering psychologist who made significant contributions to the fields of psychology and psychiatry. His groundbreaking work focused on the psychology of suggestion, dissociative identity disorder, psychopathology, and genius. He was one of the first researchers to study the effects of suggestion on behavior and mental processes, as well as the first to describe dissociative identity disorder (previously known as multiple personality disorder). He also studied the psychological aspects of genius, including its relationship to psychopathology. His work has had a lasting impact on the field of psychology and psychiatry. |