# Blaze-IT: a lightweight BERT model for the Italian language

Francesco Russo, Michele Filannino

*Prometeia S.p.A., Piazza Trento e Trieste 3, Bologna, Italy*

Abstract

In this work, we present a lightweight language model based on BERT (Blaze-IT) and a lightweight language model based on MiniLM (Flare-IT), both specifically designed for the Italian language. Starting from the multilingual cased DistilBERT and MiniLM models, we modified the embedding layers and then carried out a continued pre-training procedure on Italian Wikipedia data using whole word masking, resulting in two uncased models. Blaze-IT has 55M parameters and weighs 217MB, while Flare-IT has 17M parameters and only weighs 67MB. The models are tailored to analyze large volumes of natively digital text, such as wikis, web pages and news articles, written in correct and fluent Italian. We evaluate their performances on various downstream tasks and compare them to other models in the same class. We also discuss the limitations of our models and suggest possible directions for future work. Our results show that our models achieve competitive performances while being much smaller than other monolingual models, making them suitable for deployment in resource-constrained environments.

**Keywords**

BERT, MiniLM, Natural Language Processing, Foundation models, Italian

## 1. Introduction

Natural Language Processing (NLP) has rapidly advanced in recent years, with language models such as BERT [1] (and its variants) and GPT [2] achieving state-of-the-art results in various NLP tasks. However, the sheer size and complexity of these models pose a significant challenge when it comes to analyzing large volumes of data or deploying applications in low-resource settings, where CPU parallelization is the only viable way to speed up the computation (since GPUs are not available or not cost effective) and loading multiple models in parallel can quickly flood the RAM.

While some previous work focused on creating a small uncased model for the Italian language exploiting knowledge distillation [3] (which produced an effective Italian DistilBERT model [4], with $\sim 40\%$ less parameters than a classic BERT model), other research went on to reduce the size of the embedding layer to focus a multilingual model on a single language [5]. Quantization and pruning techniques are also widely used. [6] [7].

In this paper, we present two lightweight language models, based on BERT [1] and MiniLM [8] respectively, both designed specifically for the Italian language. The first one (Blaze-IT) is overall 50% lighter than typical mono-lingual BERT models and 20% lighter than standard DistilBERT models, while still producing high-quality results (see the section Results). The second one (Flare-IT) is 85% lighter than mono-lingual BERT and 75% lighter than DistilBERT. In addition, both models are uncased, which makes them extremely versatile and suitable for a wide spectrum of scenarios where word capitalization might not be respected or reliable.

Our models can effectively process natural language inputs and perform a wide range of NLP tasks such as topic modeling, named entity recognition and question answering, therefore highlighting the importance of developing lightweight language models that can operate effectively in resource-constrained settings, making NLP accessible to a wider range of use-cases.

## 2. Blaze-IT and Flare-IT

The first proposed language model (Blaze-IT) is based on the multilingual cased DistilBERT model [3] (distilbert-base-multilingual-cased, 6 hidden layers and hidden size of 768, developed by the HuggingFace team as a distilled version of the original multilingual BERT). To focus this model on the Italian language, we first modified the embedding layer, following the approach presented in [5], which we extended to the deletion of cased tokens to turn the original cased model into an uncased version. This was achieved by turning the Italian language subset of the Wikipedia dataset to lowercase, tokenizing the texts with the WordPiece tokenizer of the DistilBERT model, and then computing document-level frequencies of tokens, setting a minimum threshold of 0.1% to determine which tokens to keep.

The same procedure was followed with the second model (Flare-IT) except that in this case the mMiniLMv2 model [8] (L6xH384 mMiniLMv2, 6 layers and hidden size of 384, developed by Microsoft as a distilled version of XLM-RoBERTa-Large) was used as a starting point.

*CLiC-it 2023: 9th Italian Conference on Computational Linguistics, Nov 30 — Dec 02, 2023, Venice, Italy*

francesco.russo@prometeia.com (F. Russo); michele.filannino@prometeia.com (M. Filannino)

However, the resulting models were still relying on their original training, which exploited capitalized representations of several words (like words at the beginning of sentences, or proper names of people, places and other entities), so they were not properly trained to deal with the lowercase equivalents. Moreover, while many commonly capitalized words were previously represented by a single token, their lowercase equivalent is likely to be splitted in several subword tokens, being less common than the capitalized version (e.g. "Microsoft" → "micro ##so ##ft"), which makes it harder for the models to deal with them properly, especially in token classification tasks.

To make the models more robust to the lowercase representations of words previously capitalized and compensate for the deletion of cased tokens, we exploited a continued pre-training procedure [9] [10]. More specifically, we further pre-trained the models on the Italian split of the Wikipedia dataset, using the whole word masking technique [11]. By masking whole words at once, rather than individual tokens, this technique makes the Masked Language Modeling (MLM) task harder for the models, encouraging them to learn more effective representations and to capture a wider range of linguistic structures.

Overall, these modifications allowed us to adapt the two pre-existing multilingual language models to the Italian language, and to turn them from case-sensitive to case-insensitive, significantly reducing the size of the models while maintaining their ability to produce effective representations of Italian text.

Blaze-IT has 55M parameters, a vocabulary of 13.832 tokens, and a size of 217MB. Flare-IT has 17M parameters, a vocabulary of 14.610 tokens, and a size of 67MB. The models can be fine-tuned for a wide range of downstream NLP tasks, making them highly versatile and useful for practical applications (we fine-tuned them on Text Classification, Part Of Speech Tagging, Named Entity Recognition, Semantic Textual Similarity and Extractive Question Answering, reporting the results in the dedicated section Results). A short comparison [1] between Blaze-IT, Flare-IT, BERT [2] and DistilBERT [3] is summarized in Tables 1, 2, 3 and 4.

## 2.1. Training details

The proposed Italian language models have been trained using Masked Language Modeling (MLM) on the Italian subset of the Wikipedia dataset, which contains approximately 3.7GB of text data (we used a 2020 dump of Wikipedia, already pre-processed by the HuggingFace team). Specifically, adapting from the continued pre-training setups in [9] and [10], the models were trained for 10,000 steps using the `AdamW` optimizer with a batch size of 512, obtained through 128 gradient accumulation steps and an instantaneous batch size of 4 on a NVIDIA GeForce RTX 3060 GPU. We kept the sequence length fixed to 512 and applied a linearly decaying learning rate starting from $5 \cdot 10^{-5}$.

Following the original pre-training strategy of BERT [1], we masked 15% of the tokens for each training instance, where 80% are effectively replaced by a [MASK] token, 10% are replaced by a random token and 10% are left unchanged. However, unlike the original BERT training procedure, and in agreement with the improvements introduced by the RoBERTa procedure [12], we removed

**Table 1**
Comparison across the major model indicators between Blaze-IT and BERT.

| | **Blaze-IT** | **BERT** | $\Delta\%$ |
|---|---|---|---|
| Vocab | 13.832 | 32.102 | -56,9% |
| Params | 54.150.920 | 110.727.782 | -51,1% |
| Size | 217MB | 445MB | -51,2% |

**Table 2**
Comparison across the major model indicators between Flare-IT and BERT.

| | **Flare-IT** | **BERT** | $\Delta\%$ |
|---|---|---|---|
| Vocab | 14.610 | 32.102 | -54,5% |
| Params | 16.618.770 | 110.727.782 | -85,0% |
| Size | 67MB | 445MB | -84,9% |

**Table 3**
Comparison across the major model indicators between Blaze-IT and DistilBERT.

| | **Blaze-IT** | **DistilBERT** | $\Delta\%$ |
|---|---|---|---|
| Vocab | 13.832 | 32.102 | -56,9% |
| Params | 54.150.920 | 68.200.550 | -20,6% |
| Size | 217MB | 273MB | -20,5% |

**Table 4**
Comparison across the major model indicators between Flare-IT and DistilBERT.

| | **Flare-IT** | **DistilBERT** | $\Delta\%$ |
|---|---|---|---|
| Vocab | 14.610 | 32.102 | -54,5% |
| Params | 16.618.770 | 68.200.550 | -75,6% |
| Size | 67MB | 273MB | -75,5% |

[1] The MB sizes of the models are referred to their PyTorch checkpoints. Since their exact value can slightly vary depending on the platform, we used the size of the .bin files uploaded on HuggingFace as a reference

[2] We used the bert-base-italian-xxl-uncased model on HuggingFace, released by the Bavarian State Library MDZ team, as a reference BERT model

[3] We used the BERTino model on HuggingFace, released by indigo.ai, as a reference DistilBERT model

the Next Sentence Prediction task from the pre-training.

To ensure optimal performance during training, we also employed dynamic masking [12] between epochs and utilized the whole word masking technique to encourage the models to learn more effective representations of Italian lowercased text. The dynamic masking technique involves randomly masking tokens in the input sequence during training over different epochs, while the whole word masking technique involves masking entire words at once rather than just individual tokens. Together, these techniques help to prevent overfitting and improve the robustness of the models.

The resulting models have been fine-tuned and evaluated on a range of benchmark datasets, demonstrating comparable performances with other models in their class. The limited size of the models, combined with their performances, makes them highly valuable assets for large-scale data analysis, especially in resource-constrained settings or in applications where computational efficiency is a priority, without excessively compromising on output quality.

# 3. Results

The metrics in Table 5, 6, 7, 8, 9 have been computed by fine-tuning our models and the reference models on:

- `Text Classification`: XGLUE NC, machine-translated from English [4] [13]
- `Part of Speech Tagging`: UD Italian ISDT dataset [5] [14]
- `Named Entity Recognition`: WikiNER dataset [6] [15]
- `Semantic Similarity`: MULTI STS-B dataset [16]
- `Extractive Question Answering`: SQuAD-IT dataset [17]

The Text Classification models have been trained for 1 epoch and the PoST models for 5 epochs, while the NER, STS and EQA models have been trained for 3 epochs, all with a constant learning rate, fixed at $10^{-5}$. For Text Classification, Part of Speech Tagging, Semantic Similarity and Extractive Question Answering, the metrics have been computed on the default test set provided with the dataset, while for Named Entity Recognition the metrics have been computed with a 5-fold cross-validation.

For Text Classification on the NC dataset, the Accuracy metric has been used, in agreement with the original XGLUE paper. For Token Classification tasks, the Recall, Precision and F1 metrics have been computed at the token level and then macro-averaged over the classes. For STS and EQA the Pearson's $r$ and Exact Match metrics have been used, respectively.

**Table 5**
Text Classification results

| Model | Accuracy |
|---|---|
| BERT | 90.73 |
| DistilBERT | 90.17 |
| Blaze-IT | 90.13 |
| Flare-IT | 86.23 |

**Table 6**
Part of Speech Tagging results

| Model | Recall | Precision | F1 |
|---|---|---|---|
| BERT | 97.89 | 97.74 | 97.80 |
| DistilBERT | 97.64 | 97.45 | 97.53 |
| Blaze-IT | 97.48 | 97.29 | 97.37 |
| Flare-IT | 95.64 | 95.32 | 95.45 |

**Table 7**
Named Entity Recognition results

| Model | Recall | Precision | F1 |
|---|---|---|---|
| BERT | 92.04 | 91.49 | 91.75 |
| DistilBERT | 90.76 | 91.30 | 91.01 |
| Blaze-IT | 89.29 | 89.84 | 89.53 |
| Flare-IT | 82.27 | 80.64 | 81.29 |

**Table 8**
Extractive Question Answering results

| Model | EM |
|---|---|
| BERT | 61.03 |
| DistilBERT | 56.64 |
| Blaze-IT | 55.08 |
| Flare-IT | 52.83 |

## 3.1. Throughput

In order to test the improvements that can be achieved by exploiting the limited weight of Blaze-IT and Flare-IT, we conducted an experiment which simulates the typical conditions of a cloud instance. More specifically, we set up a Docker image with the relevant requirements for an inference task (we chose the Text Classification task on the NC dataset), and then launched a container with 8 CPU cores and a 8GB RAM memory budget. For each one of the models, we tried to achieve the maximum level of parallelization allowed by the RAM (i.e. the maximum

[4] We used the Helsinki-NLP/opus-mt-en-it from HuggingFace for the translation

[5] Italian corpus annotated according to the UD scheme, obtained by conversion from ISDT, released for the shared task at Evalita-2014

[6] The B-type and I-type categories have been collapsed together since the B-type categories have extremely limited support

**Table 9**
Semantic Textual Similarity results

| Model | Pearson's $r$ |
|---|---|
| BERT | 0.8234 |
| DistilBERT | 0.7920 |
| Blaze-IT | 0.7768 |
| Flare-IT | 0.7572 |

**Table 10**
Throughput measurements with a fixed memory budget

| Model | N. jobs | Samples / s | $\Delta\%$ |
|---|---|---|---|
| BERT | 1 | 1.06 | // |
| DistilBERT | 3 | 2.25 | +112% |
| Blaze-IT | 4 | 2.49 | +135% |
| Flare-IT | 8 | 5.40 | +420% |

number of parallel jobs that could be launched without getting a SIGKILL signal from the operating system)

We then proceeded to measure the throughput reached by the models, each one with its maximum parallelization, and the relative increase in throughput compared to a classical BERT model. The results are in Table 10

## 3.2. Limitations

The proposed lightweight Italian language models have been further pre-trained, in our work, on the Italian subset of the Wikipedia dataset, which consists of high-quality, natively digital text written in a correct and fluent form. As a result, the model is particularly well-suited for analyzing large volumes of text from the web, such as wikis, web pages, news articles, and other similar sources.

However, it is worth noting that the models may have limitations when it comes to analyzing chaotic text that contains errors, slang expressions, or other types of noise. This is because such text is often less structured and less consistent than the text found in more formal digital sources, which can make it more difficult for the models to accurately process and interpret. Additionally, the models may struggle when analyzing domain-specific text, such as medical, financial, or legal content, which often contains specialized terminology and conventions that may not be present in more general digital text.

Despite these limitations, the lightweight design and robust performances of the proposed models make them extremely valuable for a wide range of natural language processing applications. In particular, their efficiency and agility make them well-suited for analyzing large volumes of digital text or processing inputs in real-time, which can be useful in a variety of contexts, including intelligent document processing, conversational systems and web content analysis. Furthermore, the performances of the models on specific domains can be improved through further pre-training by incorporating additional training data, which may help to overcome some of the limitations we have mentioned.

# 4. Related work

Language models have become a crucial component of many natural language processing applications, ranging from text classification and sentiment analysis to machine translation and question answering. Recent advances in transformer-based architectures, such as BERT [1] and GPT [2], have significantly improved the state-of-the-art performance in a wide range of natural language processing tasks. However, these models are often large and computationally expensive, making them difficult to deploy in resource-constrained environments.

Indeed, large transformers are especially cumbersome to deal with when only CPUs are available as processing units, since the execution speed is going to be heavily limited and the sheer size of the neural networks makes it hard to deploy multiple models in parallel without flooding the RAM. The huge weight of these models can also become an obstacle in cloud computing, when server-less applications exploiting state-less functions are involved, since loading large models takes lots of time, which goes against the idea of executing a function on-the-fly.

To address these issues, several approaches have been proposed to reduce the complexity and size of language models without compromising their performance. One such approach is distillation [3], in which a larger, pre-trained model is used to train a smaller, "distilled" model that can achieve comparable performance. However, knowledge distillation is computationally expensive, and while several of these compressed version have been released for English models, or for multilingual models, only a few studies have focused specifically on developing lightweight language models for low-resource languages, which may also lack the large, high-quality training datasets that are available for more widely spoken languages.

Another approach is pruning, in which unimportant connections (or even entire layers) are removed from the model to reduce its size [7] [18]. While pruning techniques are effective up to a certain point, they can affect the performances when relevant fractions of the models are removed.

A different technique is quantization [6], which exploits less accurate representations of floating points (e.g. 16 bits instead of 32) so that the resulting model is lighter (even though this is only strictly true for half-precision, because working in mixed-precision with master weights will actually lead to two copies of the model weights being loaded, one in FP32 and one in FP16 [19]).

Lastly, the modification of the embedding layer proposed in [5], which is the method we followed in our work (and can be seen as a form of pruning where only weights corresponding to unused tokens are removed), allows to focus a multilingual model on a single language by getting rid of the extra parameters in the embedding layer, therefore reducing its size (the parameters in the embedding layer are a considerable portion of the total parameters when the model's vocabulary is large). This procedure implies a limited (or sometimes even negligible) loss in performance, since it only affects statistically rare tokens for the target language.

When applied to the distilled versions of multilingual models, this technique can further reduce their size and, as we showed in our work, if cased tokens are also removed (with the inclusion of an additional pre-training phase to compensate for their deletion, possibly exploiting whole word masking) the procedure ultimately delivers extremely light language models, with the additional benefit of the new uncased representations.

The introduction of an additional pre-training phase, formally known as continued pre-training, has been mainly explored as a method to adapt language models to new domains or tasks [10] ("domain-adaptive pre-training" and "task-adaptive pre-training"), yielding improvements in downstream performances on Text Classification. Recent work on Spoken Language Understanding [9] has brought this further by investigating the effectiveness of the continued pre-training of English models in cross-lingual settings, showing that the domain knowledge obtained on intermediate data is even transferable to other languages.

In our work, we exploited continued pre-training to adapt our language models to uncased text, taking inspiration from the setups in [9] and [10].

## 5. Conclusions

In this paper, we presented Blaze-IT, a lightweight language model based on BERT, and Flare-IT, a lightweight language model based on MiniLM, both specifically tailored for the Italian language. Our models are significantly smaller than other monolingual Italian models, the first one weighing 217MB and having 55M parameters, the second one weighing only 67MB and having 17M parameters. We achieved this by starting with the multilingual DistilBERT and MiniLM models, reducing the embedding layer and further pre-training them on Italian Wikipedia data using whole word masking. While the models are designed to excel on correctly written digital text, they may struggle with noisy, informal language or domain-specific jargon.

The limited size of our models makes them well-suited for local environments and applications where large volumes of data have to be processed, especially if no hardware acceleration is available, since the execution of these light models can be easily parallelized on multiple CPUs. They are also ideal when computational resources or memory are limited, such as on mobile devices or edge-computing environments, or even in cloud-computing scenarios where server-less applications are involved, since these models can be quickly loaded and used in state-less functions. We hope that our work will help to lower the entry barrier for natural language processing tasks for researchers and practitioners working in low-resource settings.

In future work, we plan to investigate methods for further compressing the size of transformer-based models while maintaining performances, perhaps combining the techniques showed in this work with model quantization. We also aim to expand the capabilities of our models, to handle informal and noisy text, and to develop domain-specific versions of the models for specialized applications. Overall, we believe that this work represents a step towards democratizing access to natural language processing tools and techniques, and we look forward to further developments in this area.

You can find the models online on the HuggingFace platform at https://huggingface.co/osiria/blaze-it and https://huggingface.co/osiria/flare-it

You can also try the models online (fine-tuned on named entity recognition) using the web apps at https://huggingface.co/spaces/osiria/blaze-it-demo and https://huggingface.co/spaces/osiria/flare-it-demo.

Blaze-IT is released under Apache-2.0 license and Flare-IT under MIT license.

## 6. Acknowledgments

The authors would like to acknowledge Prometeia S.p.A. for the unconditioned support, availability and guidance received. The authors would also like to thank Alessio Palmero Aprosio (FBK) for his valuable suggestions and feedback.

## References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen,

E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.

[3] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).

[4] M. Muffo, E. Bertino, Bertino: An italian distilbert model, ArXiv abs/2303.18121 (2023).

[5] A. Abdaoui, C. Pradel, G. Sigel, Load what you need: Smaller versions of mutlilingual bert, in: SUSTAINLP, 2020.

[6] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, K. Keutzer, A survey of quantization methods for efficient neural network inference, ArXiv abs/2103.13630 (2021).

[7] M. A. Gordon, K. Duh, N. Andrews, Compressing bert: Studying the effects of weight pruning on transfer learning, ArXiv (2020).

[8] W. Wang, H. Bao, S. Huang, L. Dong, F. Wei, Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers, in: Findings, 2020.

[9] S. M. B. Louvan, Samuel; Casola, Investigating continued pretraining for zero-shot cross-lingual spoken language understanding, in: Proceedings of the Eighth Italian Conference on Computational Linguistics Lingua/e Inglese, 2018.

[10] S. Gururangan, A. Marasovi'c, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020.

[11] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, G. Hu, Pre-training with whole word masking for chinese bert, ArXiv abs/1906.08101 (2019).

[12] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, 2021.

[13] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, X. Fan, R. Zhang, R. Agrawal, E. Cui, S. Wei, T. Bharti, Y. Qiao, J.-H. Chen, W. Wu, S. Liu, F. Yang, D. Campos, R. Majumder, M. Zhou, XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2020.

[14] C. Bosco, F. Dell'Orletta, S. Montemagni, M. Sanguinetti, M. Simi, The evalita 2014 dependency parsing task, Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 and of the Fourth International Workshop (2014).

[15] J. Nothman, N. Ringland, W. Radford, T. Murphy, J. R. Curran, Learning multilingual named entity recognition from wikipedia, Artificial Intelligence 194 (2013).

[16] P. May, Machine translated multilingual sts benchmark dataset., 2021. URL: https://github.com/PhilipMay/stsb-multi-mt.

[17] D. Croce, A. Zelenanska, R. Basili, Neural learning for question answering in italian, in: International Conference of the Italian Association for Artificial Intelligence, 2018.

[18] H. Sajjad, F. Dalvi, N. Durrani, P. Nakov, On the effect of dropping layers of pre-trained transformer models, Computer Speech & Language (2023).

[19] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, H. Wu, Mixed precision training, in: International Conference on Learning Representations, 2018.