

BERTino: an Italian DistilBERT model

Matteo Muffo

Indigo.ai
Via Torino 61, Milano
matteo@indigo.ai

Enrico Bertino

Indigo.ai
Via Torino 61, Milano
e@indigo.ai

Abstract

English.¹ The recent introduction of Transformers language representation models allowed great improvements in many natural language processing (NLP) tasks. However, if on one hand the performances achieved by this kind of architectures are surprising, on the other their usability is limited by the high number of parameters which constitute their network, resulting in high computational and memory demands. In this work we present BERTino, a DistilBERT model which proposes to be the first lightweight alternative to the BERT architecture specific for the Italian language. We evaluated BERTino on the Italian ISDT, Italian ParTUT, Italian WikiNER and multiclass classification tasks, obtaining F1 scores comparable to those obtained by a $BERT_{BASE}$ with a remarkable improvement in training and inference speed.

Italiano. La recente introduzione dei Transformers come modelli di rappresentazione del linguaggio naturale ha permesso grandi avanzamenti sullo stato dell'arte in molte applicazioni di Natural Language Processing (NLP). Tuttavia, se da una parte i risultati raggiunti da queste architetture sono sorprendenti, dall'altra la loro fruibilità è limitata dall'elevato numero di parametri che costituiscono la loro architettura, con conseguenti elevate esigenze computazionali e di memoria. In questo lavoro presentiamo BERTino, un modello DistilBERT che è la prima

alternativa *leggera* all'architettura BERT specifica per la lingua italiana. Abbiamo valutato BERTino sui task ISDT italiano, ParTUT italiano, WikiNER italiano e classificazione multiclasse, ottenendo punteggi F1 paragonabili a quelli ottenuti da un modello $BERT_{BASE}$ con un notevole miglioramento nella velocità di addestramento e inferenza.

1 Introduction

In recent years the introduction of Transformers language models allowed great improvements in many natural language processing (NLP) tasks. Among Transformer language models, BERT (Devlin et al., 2018) affirmed itself as a high-performing and flexible alternative, being able to transfer knowledge from general tasks to downstream ones thanks to the pretraining-finetuning approach. The context-dependent text representations provided by this model demonstrated to be a richer source of information when compared to static textual embeddings such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2016) or Sent2Vec (Pagliardini et al., 2018). However, despite the substantial improvements brought by BERT in the NLP field, the high number of parameters that constitute its network makes its usage prohibitive in resource-limited devices, both at training and inference time, and with a non-negligible environmental impact. To address the aforementioned problem, recent research proposes several approaches to reduce the size of the BERT network, such as DistilBERT (Sanh et al., 2019), MobileBERT (Sun et al., 2020) or pruning (Gordon et al., 2020; McCarley et al., 2019).

The experiments conducted in Virtanen et al. (2019), de Vries et al. (2019) and Martin et al. (2020) demonstrate that monolingual BERT models outperform the same multilingual BERT ar-

¹Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

chitecture (Devlin et al., 2018), justifying the effort for pre-training Transformer models required for specific languages. In this work we present **BERTino**, a DistilBERT model pre-trained on a large Italian corpus. This model proposes to be the first general-domain, lightweight alternative to BERT specific for the Italian language. We evaluate BERTino on two Part Of Speech tagging tasks, Italian ISDT (Bosco et al., 2000) and Italian PARTUT (Sanguinetti and Bosco, 2015), on the Italian WikiNER (Nothman et al., 2012) Named Entity Recognition task and on a multi-class sentence classification. Comparing the scores obtained by BERTino, its teacher model and GilBERTo, the first obtains performances comparable to the other two architectures while sensibly decreasing the fine-tuning and evaluation time. In Section 2 we discuss the related works with a focus on DistilBERT, in Section 3 we describe the corpus and the pre-train followed by the results in Section 4.

2 Related work

In this section we will give a brief outline of the inner workings for Transformers, then we overview some lightweight alternatives to BERT.

The introduction of Transformer blocks (Vaswani et al., 2017) in language representation models is a keystone in recent NLP. The attention mechanism adopted by the Transformer encoder allows to provide contextualized representations of words, which proved to be a richer source of information than static word embeddings. Attention mechanism processes all words in an input sentence simultaneously, allowing parallelization of computations. This is a non-negligible improvement with respect to models like ELMo (Peters et al., 2018), which aim to provide contextualized text representations using a bidirectional LSTM network, processing each word sequentially.

Among language models that adopt Transformer technology, BERT (Devlin et al., 2018) affirmed itself as a flexible and powerful alternative, being able to establish new state-of-the-art for 11 NLP tasks at the time of publication. In its base version, this model adopts an hidden size of 768 and is composed of 12 layers (Transformer blocks), each of these involving 12 attention heads, for a total of 110 millions of parameters. As outlined in Section 1, the high number of parameters constituting BERT’s network can result prohibitive for deployment in resource-

limited devices and the computational effort is not negligible. For this reason, great effort has been devoted by researchers in order to propose smaller but valid alternatives to the base version of BERT. Gordon et al. (2020) studies how weight pruning affects the performances of BERT, concluding that a low level of pruning (30-40% of weights) marginally affects the natural language understanding capabilities of the network.

McCarley et al. (2019) conducts a similar study on BERT weight pruning, but applied to the Question Answering downstream task specifically.

Sanh et al. (2019) propose DistilBERT, a smaller BERT architecture which is trained using the knowledge distillation technique (Hinton et al., 2015). Since the model that we propose relies on this training technique, we propose a brief description of knowledge distillation in section 2.1. DistilBERT leverages the inductive biases learned by larger models during pre-training using a triple loss combining language modeling, distillation and cosine-distance losses. DistilBERT architecture counts 40% less parameters but is able to retain 97% of natural language understanding performances with respect to the teacher model, while being 60% faster.

Sun et al. (2020) propose MobileBERT, a compressed BERT model which aims to reduce the hidden size instead of the depth of the network. As DistilBERT, MobileBERT uses knowledge distillation during pre-training but adopts a $BERT_{LARGE}$ model with inverted bottleneck as teacher.

2.1 Knowledge distillation

Knowledge distillation (Hinton et al., 2015) is a training technique that leverages the outputs of a big network (called *teacher*) to train a smaller network (the *student*). In general, in the context of supervised learning, a classifier is trained in such a way that the output probability distribution that it provides is as similar as possible to the one-hot vector representing the gold label, by minimizing the cross-entropy loss between the two. By receiving a one-hot vector as learning signal, a model evaluated on the training set will provide an output distribution with a near-one value in correspondence of the right class, and all near-zero values for other classes. Some of the near-zero probabilities, however, are larger than the others and are the result of the generalization capabili-

ties of the model. The idea of knowledge distillation is to substitute the usual one-hot vector representing gold labels with the output distribution of the teacher model in the computation of the cross-entropy loss, in order to leverage the information contained in the near-zero values of the teacher’s output distribution. Formally, the knowledge distillation loss is computed as:

$$\mathcal{L}_{KD} = \sum_i t_i * \log(s_i) \quad (1)$$

with t_i being the output distribution of the teacher model relative to the i^{th} observation, and s_i being the output distribution of the student model relative to the i^{th} observation.

3 BERTino

As outlined in section 1, we propose in this work BERTino, a DistilBERT model pre-trained on a general-domain Italian corpus. As for BERT-like architectures, BERTino is task-agnostic and can be fine-tuned for every downstream task. In this section we will report details relative to the pre-training that we conducted.

3.1 Corpus

The corpus that we used to pre-train BERTino is the union of PAISA (Lyding et al., 2014) and ItWaC (Baroni et al., 2009), two general-domain Italian corpora scraped from the web. While the former is made up of short sentences, the latter includes a considerable amount of long sentences. Since our model can receive input sequences of at most 512 tokens, as for BERT architectures, we decided to apply a pre-processing scheme to the ItWaC corpus. We split the sentences with more than 400 words into sub-sentences, using fixed points to create chunks that keep the semantic sense of a sentence. In this way, most of the long sentences contained in ItWaC are split into sub-sentences containing less than 512 tokens. A certain number of the final sentences still contain more than 512 tokens and they will be useful for training the parameters relative to the last entries of the network.

The PAISA corpus counts 7.5 million sentences and 223.5 million words. The ItWaC corpus counts 6.5 million sentences and 1.6 billion words after preprocessing. Our final corpus counts 14 million sentences and 1.9 billion words for a total of 12GB of text.

3.2 Pre-training

Teacher model The teacher model that we selected to perform knowledge distillation during the pre-training of BERTino is *dbmdz/bert-base-italian-xxl-uncased*, made by *Bavarian State Library*². We chose this model because it is the Italian $BERT_{BASE}$ model trained on the biggest corpus (81 GB of text), up to our knowledge. Following Sanh et al. (2019), we initialized the weights of our student model by taking one layer out of two from the teacher model.

Loss function We report the loss function used to pre-train BERTino:

$$\mathcal{L} = 0.45\mathcal{L}_{KD} + 0.45\mathcal{L}_{MLM} + 0.1\mathcal{L}_{COS} \quad (2)$$

with \mathcal{L}_{KD} being the knowledge distillation loss as described in equation 1, \mathcal{L}_{MLM} being the masked language modeling loss and \mathcal{L}_{COS} being the cosine embedding loss. Sanh et al. (2019) describe the cosine embedding loss useful to “align the directions of the student and teacher hidden states vectors”. When choosing the weights of the three loss functions, we wanted our model to learn from the teacher and by itself in an equal way, so we set the same weights for both \mathcal{L}_{KD} and \mathcal{L}_{MLM} . Moreover, we considered the alignment of student and teacher hidden states vectors marginal for our objective, setting \mathcal{L}_{COS} as 10% of the total loss.

Architecture The architecture of BERTino is the same as in DistilBERT. Our model adopts an hidden size of 768 and is composed of 6 layers (Transformer blocks), each of which involving 12 attention heads. In this way BERTino’s network results to have half the layers present in the $BERT_{BASE}$ architecture.

Training details To pre-train BERTino we used a batch size of 6 and an initial learning rate of 5×10^{-4} , adopting Adam (Kingma and Ba, 2014) as optimization algorithm. We chose 6 as batch size due to the limited computational resources available. Results described in section 4 demonstrate that the small batch size that we adopted is sufficient to obtain a valid pre-trained model. We trained our model on 4 Tesla K80 GPUs for 3 epochs, requiring 45 days of computation in total. For some aspects of the training, we relied on the Huggingface Transformers repository (Wolf et al., 2019).

²<https://github.com/dbmdz/berts>

4 Results

We tested the performances of BERTino on benchmark datasets: the Italian ISDT (Bosco et al., 2000) and Italian ParTUT (Sanguinetti and Bosco, 2015) Part Of Speech tagging tasks, and the Italian WikiNER (Nothman et al., 2012) Named Entity Recognition task. To complete the evaluation of the model, we also tested it on a multi-class sentence classification task. In particular, we focused on intent detection, a task specific to the context of Dialogue Systems, creating a novel Italian dataset which is freely available at our repository³. The dataset that we propose collects 2786 real-world questions (2228 for training and 558 for testing) submitted to a digital conversational agent. The total number of classes in the dataset is 139.

For the first two tasks mentioned, we fine-tuned our model on the training set for 4 epochs with a batch size of 32 and a learning rate of 5×10^{-5} , for the NER task we performed 5-fold splitting of the dataset and fine-tuned BERTino for 2 epochs per fold with a batch size of 32 and a learning rate of 5×10^{-5} , while for the multi-class classification task we fine-tuned our model for 14 epochs on the training set with a batch size of 32 and a learning rate of 5×10^{-5} . To compare the results obtained, we fine-tuned the teacher model and a GilBERTo model⁴ on the same tasks with the same hyperparameters. Tables 1, 2, 3 and 4 collect the F1 scores gathered in these experiments together with fine-tuning and evaluation time. All the scores reported represent the average computed over three different runs. Results show that the teacher model slightly outperforms BERTino, with an increase of the F1 score of 0,29%, 5,15%, 1,37% and 1,88% over the tasks analysed. However BERTino results to be a sensibly faster network with respect to the teacher model and GilBERTo, taking almost half of the time to perform both fine-tuning and evaluation. We can conclude from the last observation that BERTino is able to retain most of the natural language understanding capabilities of the teacher model, even with a much smaller architecture.

5 Conclusions

In this work we presented BERTino, a DistilBERT model which aims to be the first lightweight alternative to BERT specific for the Italian language. Our model has been trained on a general-domain

corpus and can then be finetuned with good performances on a wide range of tasks like its larger counterparts. BERTino showed comparable performances with respect to both the teacher model and GilBERTo in the Italian ISDT, Italian ParTUT, Italian WikiNER and multi-class sentence classification tasks while taking almost half of the time to fine-tune, demonstrating to be a valid lightweight alternative to $BERT_{BASE}$ models for the Italian language.

³<https://github.com/indigo-ai/BERTino>

⁴Available at <https://github.com/idb-ita/GilBERTo>

Italian ISDT			
Model	F1 score	Fine-tuning time	Evaluation time
BERTino	0,9800	9'10"	3"
Teacher model	0,9829	16'32"	6"
GilBERTo	0,9804	18'11"	5"

Table 1: F1 scores obtained by BERTino and the teacher model in the Italian ISDT task.

Italian ParTUT			
Model	F1 score	Fine-tuning time	Evaluation time
BERTino	0,9193	1'19"	1"
Teacher model	0,9708	2'19"	1"
GilBERTo	0,9621	2'21"	1"

Table 2: F1 scores obtained by BERTino and the teacher model in the Italian ParTUT task.

Italian WikiNER			
Model	F1 score	Fine-tuning time	Evaluation time
BERTino	0,9039	38'3"	3'2"
Teacher model	0,9176	67'2"	5'21"
GilBERTo	0,9136	66'33"	5'9"

Table 3: F1 scores obtained by BERTino and the teacher model in the Italian WikiNER task. The results reported are the average of the scores obtained in each of the 5 folds.

Multi-class sentence classification			
Model	F1 score	Fine-tuning time	Evaluation time
BERTino	0,7766	5'4"	6"
Teacher model	0,7954	9'48"	10"
GilBERTo	0,7381	10'0"	10"

Table 4: F1 scores obtained by BERTino and the teacher model in the multi-class sentence classification task.

References

- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information.
- Cristina Bosco, Vincenzo Lombardo, Daniela Vassallo, and Leonardo Lesmo. 2000. Building a treebank for Italian: a data-driven annotation schema. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, May. European Language Resources Association (ELRA).
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.
- Mitchell A. Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.
- Verena Lyding, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell’Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. 2014. The PAISÀ corpus of Italian web texts. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 36–43, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- J. S. McCarley, Rishav Chakravarti, and Avirup Sil. 2019. Structured pruning of a bert-based question answering model.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.
- Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2012. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.
- Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.
- Manuela Sanguinetti and Cristina Bosco, 2015. *Part-TUT: The Turin University Parallel Treebank*, pages 51–69. Springer International Publishing, Cham.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.
- Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.