ALBERTO: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets

Marco Polignano
University of Bari A. Moro
Dept. Computer Science
E.Orabona 4, Italy
marco.polignano@uniba.it

Pierpaolo Basile
University of Bari A. Moro
Dept. Computer Science
E.Orabona 4, Italy
pierpaolo.basile@uniba.it

Marco de Gemmis
University of Bari A. Moro
Dept. Computer Science
E.Orabona 4, Italy
marco.degemmis@uniba.it

Giovanni Semeraro
University of Bari A. Moro
Dept. Computer Science
E.Orabona 4, Italy
giovanni.semeraro@uniba.it

Valerio Basile
University of Turin
Dept. Computer Science
Via Verdi 8, Italy
valerio.basile@unito.it

Abstract

English. Recent scientific studies on natural language processing (NLP) report the outstanding effectiveness observed in the use of context-dependent and task-free language understanding models such as ELMo, GPT, and BERT. Specifically, they have proved to achieve state of the art performance in numerous complex NLP tasks such as question answering and sentiment analysis in the English language. Following the great popularity and effectiveness that these models are gaining in the scientific community, we trained a BERT language understanding model for the Italian language (**AlBERTo**). In particular, AlBERTo is focused on the language used in social networks, specifically on Twitter. To demonstrate its robustness, we evaluated AlBERTo on the EVALITA 2016 task SENTIPOLC (SENTIment POLarity Classification) obtaining state of the art results in subjectivity, polarity and irony detection on Italian tweets. The pre-trained AlBERTo model will be publicly distributed through the GitHub platform at the following web address: https://github.com/marcopoli/AlBERTo-it in order to facilitate future research.

1 Introduction

The recent spread of pre-trained text representation models has enabled important progress in Natural Language Processing. In particular, numerous tasks such as part of speech tagging, question answering, machine translation, and text classification have obtained significant contributions in terms of performance through the use of distributional semantics techniques such as word embedding. Mikolov et al. (2013) notably contributed to the genesis of numerous strategies for representing terms based on the idea that semantically related terms have a similar vector representations. Such technologies as Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and FastText (Bojanowski et al., 2017) suffer from a problem that multiple concepts, associated with the same term, are not represented by different wordembedding vectors in the distributional space (context-free). New strategies such as ELMo (Peters et al., 2018), GPT/GPT-2 (Radford et al., 2019), and BERT (Devlin et al., 2019) overcome this limit by learning a language understanding model for a contextual and task-independent representation of terms. In their multilingual version, they mainly use a mix of text obtained from large corpora in different languages to build a general language model to be reused for every application in any language. As reported by the BERT documentation "the Multilingual model is somewhat worse than a single-language model. However, it is not feasible for us to train and maintain dozens of single-language model." This entails significant limitations related to the type of language learned (with respect to the document style) and the size of the vocabulary. These reasons have led us to create the equivalent of the BERT model for the Italian language and specifically on the language style used on Twitter: **AlBERTo**. This idea was supported by the intuition that many of the NLP

tasks for the Italian language are carried out for the analysis of social media data, both in business and research contexts.

2 Related Work

A Task-Independent Sentence Understanding Model is based on the idea of creating a deep learning architecture, particularly an encoder and a decoder, so that the encoding level can be used in more than one NLP task. In this way, it is possible to obtain a decoding level with weights optimized for the specific task (fine-tuning). A general-purpose encoder should, therefore, be able to provide an efficient representation of the terms, their position in the sentence, context, grammatical structure of the sentence, semantics of the terms. One of the first systems able to satisfy these requirements was ELMo (Peters et al., 2018) based on a large neural network biLSTM (2 biLSTM layers with 4096 units and 512 dimension projections and a residual connection from the first to the second layer) trained for 10 epochs on the 1B WordBenchmark (Chelba et al., 2013). The goal of the network was to predict the same starting sentence in the same initial language (like an autoencoder). It has guaranteed the correct management of polysemy of terms by demonstrating its efficacy on six different NLP tasks for which it obtained state-of-the-art results: Question Answering, Textual Entailment, Semantic Role labeling, Coreference Resolution, Name Entity Extraction, and Sentiment Analysis. Following the basic idea of ELMo, another language model called GPT has been developed in order to improve the performance on the tasks included in the GLUE benchmark (Wang et al., 2018). GPT replaces the biLSTM network with a Transformer architecture (Vaswani et al., 2017). A Transformer is an encoder-decoder architecture that is mainly based on feed-forward and multi-head attention layers. Moreover, in Transformers terms are provided as input without a specific order and consequently a positional vector is added to the term embeddings. Unlike ELMo, in GPT, for each new task, the weights of all levels of the network are optimized, and the complexity of the network (in terms of parameters) remains almost constant. Moreover, during the learning phase, the network does not limit itself to work on a single sentence but it splits the text into spans to improve the predictive capacity and the generalization power of the network. The deep neural network used is a 12-layer decoder-only transformer with masked self-attention heads (768 dimensional states and 12 attention heads) trained for 100 epochs on the BooksCorpus dataset (Zhu et al., 2015). This strategy proved to be successful compared to the results obtained by ELMo on the same NLP tasks. BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) was developed to work with a strategy very similar to GPT. In its basic version, it is also trained on a Transformer network with 12 levels, 768 dimensional states and 12 heads of attention for a total of 110M of parameters and trained on BooksCorpus (Zhu et al., 2015) and Wikipedia English for 1M of steps. The main difference is that the learning phase is performed by scanning the span of text in both directions, from left to right and from right to left, as was already done in biLSTMs. Moreover, BERT uses a "masked language model": during the training, random terms are masked in order to be predicted by the net. Jointly, the network is also designed to potentially learn the next span of text from the one given in input. These variations on the GPT model allow BERT to be the current state of the art language understanding model. Larger versions of BERT (BERT large) and GPT (GPT-2) have been released and are scoring better results than the normal scale models but require much more computational power. The base BERT model for English language is exactly the same used for learning the Italian Language Understang Model (AlBERTo) but we are considering the possibility to develop a large version of it soon.

3 AlBERTo

As pointed out in the previous sections, the aim of this work is to create a linguistic resource for Italian that would follow the most recent strategies used to address NLP problems in English. It is well known that the language used on social networks is different from the formal one, also as a consequence of the presence of mentions, uncommon terms, links, and hashtags that are not present elsewhere. Moreover multiple language models in their multilingual version, are not performing well in every specific language, especially with a writing style different from that of books and encyclopedic descriptions (Polignano et al., 2019). AlBERTo aims to be the first Italian language under-

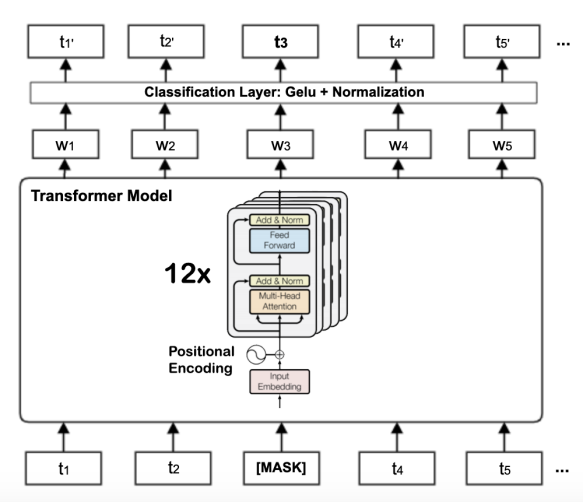

Figure 1: BERT and AlBERTo learning strategy

standing model to represent the social media language, Twitter in particular, written in Italian. The model proposed in this work is based on the software distributed through GitHub by Devlin et al. (2019) [1] with the endorsement of Google. It has been trained, without consequences, on text spans containing typical social media characters including emojis, links hashtags and mentions.

Figure 1 shows the BERT and AlBERTo strategy of learning. The "masked learning" is applied on a $12x$ Transformer Encoder, where, for each input, a percentage of terms is hidden and then predicted for optimizing network weights in back-propagation. In AlBERTo, we implement only the "masked learning" strategy, excluding the step based on "next following sentence". This is a crucial aspect to be aware of because, in the case of tweets, we do not have cognition of a flow of tweets as it happens in a dialog. For this reason, we are aware that AlBERTo is not suitable for the task of question answering, where this property is essential. On the contrary, the model is well suited for classification and prediction tasks. The decision to train AlBERTo, excluding the "next following sentence" strategy, makes the model similar in purposes to ELMo. Differently from it, BERT and AlBERTo use transformer architecture instead on biLSTM which have been demonstrated to perform better in natural language processing tasks. In any case, we are considering the possibility to learn an Italian ELMo model and to compare it with the here proposed model.

[1] https://github.com/google-research/bert/

Original tweet: *#labuonascuola Eccolo, il rapporto on line qui http://t.co/U5AXNySoJu*

Preprocessed: <hashtag> la buona scuola </hashtag> eccolo il rapporto on line qui <url>

Figure 2: Example of preprocessed Tweet

3.1 Text Preprocessing

In order to tailor the tweet text to BERT's input structure, it is necessary to carry out preprocessing operations. More specifically, using Python as the programming language, two libraries were mainly adopted: Ekphrasis (Baziotis et al., 2017) and SentencePiece[2] (Kudo, 2018). Ekphrasis is a popular tool comprising an NLP pipeline for text extracted from Twitter. It has been used for:

- Normalizing URL, emails, mentions, percents, money, time, date, phone numbers, numbers, emoticons;
- Tagging and unpacking hashtags.

The normalization phase consists in replacing each term with a fixed tuple $< [entity \quad type] >$. The tagging phase consists of enclosing hashtags with two tags $< hashtag > ... < /hashtag >$ representing their beginning and end in the sentence. Whenever possible, the hashtag has been unpacked into known words. The text is cleaned and made easily readable by the network by converting it to its lowercase form and all characters except emojis, !, ? and accented characters have been deleted. An example of preprocessed tweet is shown in Figure 2.

SentencePiece is a segmentation algorithm used for learning the best strategy for splitting text into terms in an unsupervised and language-independent way. It can process up to 50k sentences per seconds and generate an extensive vocabulary. It includes the most common terms in the training set and the subwords which occur in the middle of words, annotating them with '##' in order to be able to encode also slang, incomplete or uncommon words. An example of a piece of the vocabulary generated for AlBERTo is shown in Figure 3. SentencePiece also produced a tokenizer, used to generate a list of tokens for each tweet further processed by BERT's *create_pretraining_data.py* module.

[2] https://github.com/google/sentencepiece

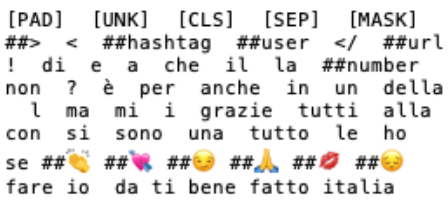

Figure 3: An extract of the vocabulary created by SentencePiece for AlBERTo

3.2 Dataset

The dataset used for the learning phase of AlBERTo is TWITA (Basile et al., 2018) a huge corpus of Tweets in the Italian language collected from February 2012 to the present day from Twitter's official streaming API. In our configuration, we randomly selected 200 million Tweets removing re-tweets, and processed them with the preprocessing pipeline described previously. In total, we obtained 191GB of raw data.

3.3 Learning Configuration

The AlBERTo model has been trained using the following configuration:

```
bert_base_config = {
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "max_position_embeddings": 512,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 128000
}
```

```
# Input data pipeline config
TRAIN_BATCH_SIZE = 128
MAX_PREDICTIONS = 20
MAX_SEQ_LENGTH = 128
MASKED_LM_PROB = 0.15

# Training procedure config
EVAL_BATCH_SIZE = 64
LEARNING_RATE = 2e-5
TRAIN_STEPS = 1000000
SAVE_CHECKPOINTS_STEPS = 2500
NUM_TPU_CORES = 8
```

The training has been performed over the Google Collaborative Environment (Colab)[3], Using a 8 core Google TPU-V2[4] and a Google Cloud Storage Bucket[5]. In total, it took $\sim$ 50 hours to create a complete AlBERTo model. More technical details are available in the Notebook *"Italian Pre-training BERT from scratch with cloud TPU"* into the project repository.

4 Evaluation and Discussion of Results

We evaluate AlBERTo on a task of sentiment analysis for the Italian language. In particular, we decided to use the data released for the SENTIPOLC (SENTIment Polarity Classification) shared task (Barbieri et al., 2016) carried out at EVALITA 2016 (Basile et al., 2016) whose tweets comes from a distribution different from them used for training AlBERTo. It includes three subtasks:

- **Subjectivity Classification**: "a system must decide whether a given message is subjective or objective";
- **Polarity Classification**: "a system must decide whether a given message is of positive, negative, neutral or mixed sentiment";
- **Irony Detection**: "a system must decide whether a given message is ironic or not".

Data provided for training and test are tagged with six fields containing values related to manual annotation: subj, opos, oneg, iro, lpos, lneg. These labels describe consequently if the sentence is subjective, positive, negative, ironical, literal positive, literal negative. For each of these classes, there is a 1 where the sentence satisfy the label, a 0 instead.
The last two labels "lpos" and "lneg" that describe the literal polarity of the tweet have not been considered in the current evaluation (nor in the official shared task evaluation). In total, 7410 tweets have been released for training and 2000 for testing. We do not used any validation set because we do not performed any phase of model selection during the fine-tuning of AlBERTo. The evaluation was performed considering precision (p), recall (r) and F1-score (F1) for each class and for each classification task.

[3] https://colab.research.google.com
[4] https://cloud.google.com/tpu/
[5] https://cloud.google.com/storage/

	Prec. 0	Rec. 0	F1. 0
Subjectivity	0.6838	0.8058	0.7398
Polarity Pos.	0.9262	0.8301	0.8755
Polarity Neg.	0.7537	0.9179	0.8277
Irony	0.9001	0.9853	0.9408
	Prec. 1	**Rec. 1**	**F1 . 1**
Subjectivity	0.8857	0.8015	0.8415
Polarity Pos.	0.5818	0.5314	0.5554
Polarity Neg.	0.7988	0.5208	0.6305
Irony	0.6176	0.1787	0.2772

Table 1: Results obtained using the official evaluation script of SENTIPOLC 2016

System	*Obj*	*Subj*	*F*
AlBERTo	***0.7398***	***0.8415***	***0.7906***
Unitor.1.u	0.6784	0.8105	0.7444
Unitor.2.u	0.6723	0.7979	0.7351
samskara.1.c	0.6555	0.7814	0.7184
ItaliaNLP.2.c	0.6733	0.7535	0.7134

System	Pos	Neg	F
AlBERTo	***0.7155***	***0.7291***	***0.7223***
UniPI.2.c	0.6850	0.6426	0.6638
Unitor.1.u	0.6354	0.6885	0.6620
Unitor.2.u	0.6312	0.6838	0.6575
ItaliaNLP.1.c	0.6265	0.6743	0.6504

System	Non-Iro	Iro	F
AlBERTo	***0.9408***	***0.2772***	***0.6090***
tweet2check16.c	0.9115	0.1710	0.5412
CoMoDI.c	0.8993	0.1509	0.5251
tweet2check14.c	0.9166	0.1159	0.5162
IRADABE.2.c	0.9241	0.1026	0.5133

Table 2: Comparison of results with the best systems of SENTIPOLC for each classification task

AlBERTo fine-tuning. We fine-tuned AlBERTo four different times, in order to obtain one classifier for each task except for the polarity where we have two of them. In particular, we created one classifier for the Subjectivity Classification, one for Polarity Positive, one for Polarity Negative and one for the Irony Detection. Each time we have re-trained the model for three epochs, using a learning rate of 2e-5 with 1000 steps per loops on batches of 512 example from the training set of the specific task. For the fine-tuning of the Irony Detection classifier, we increased the number of epochs of training to ten observing low performances using only three epochs as for the other classification tasks. The fine-tuning process lasted $\sim$ 4 minutes every time.

Discussion of the results. The results reported in Table 1 show the output obtained from the official evaluation script of SENTIPOLC 2016. It is important to note that the values on the individual classes of precision, recall and, F1 are not compared with them of the systems that participated in the competition because they are not reported in the overview paper of the task. Nevertheless, some considerations can be drawn. The classifier based on AlBERTo achieves, on average, high recall on class 0 and low values on class 1. The opposite situation is instead observed on the precision, where for the class 1 it is on average superior to the recall values. This note suggests that the system is very good at classifying a phenomenon and when it does, it is sure of the prediction made even at the cost of generating false negatives.

On each of the sub-tasks of SENTIPOLC, it can be observed that AlBERTo has obtained state of the art results without any heuristic tuning of learning parameters (model as it is after fine-tuning training) except in the case of irony detection where it was necessary to increase the number of epochs of the learning phase of fine-tuning. Comparing AlBERTo with the best system of each subtask, we observe an increase in results between 7% and 11%. The results obtained are exciting, from our point of view, for further future work.

5 Conclusion

In this work, we described AlBERTo, the first Italian language understanding model based on social media writing style. The model has been trained using the official BERT source code on a Google TPU-V2 on 200M tweets in the Italian language. The pre-trained model has been fine-tuned on the data available for the classification task SENTIPOLC 2016, showing SOTA results. The results allow us to promote AlBERTo as the starting point for future research in this direction. Model repository: https://github.com/marcopoli/AlBERTo-it

Acknowledgment

The work of Marco Polignano is funded by project "DECiSION" codice raggruppamento: BQS5153, under the Apulian INNONETWORK programme, Italy. The work of Valerio Basile is partially funded by Progetto di Ateneo/CSP 2016 (*Immigrants, Hate and Prejudice in Social Media*, S1618_L2_BOSC_01.

References

Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. 2016. Overview of the evalita 2016 sentiment polarity classification task. In *Proceedings of third Italian conference on computational linguistics (CLiC-it 2016) & fifth evaluation campaign of natural language processing and speech tools for Italian. Final Workshop (EVALITA 2016)*.

Pierpaolo Basile, Franco Cutugno, Malvina Nissim, Viviana Patti, Rachele Sprugnoli, et al. 2016. Evalita 2016: Overview of the 5th evaluation campaign of natural language processing and speech tools for italian. In *3rd Italian Conference on Computational Linguistics, CLiC-it 2016 and 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, EVALITA 2016*, volume 1749, pages 1–4. CEUR-WS.

Valerio Basile, Mirko Lai, and Manuela Sanguinetti. 2018. Long-term social media data collection at the university of turin. In *Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, pages 1–6. CEUR-WS.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

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, June. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. pages 2227–2237, June.

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2019. A comparison of word-embeddings in emotion detection from text using bilstm, cnn and self-attention. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 63–68. ACM.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

https://github.com/marcopoli/AlBERTo-it