ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI FORLÌ

91258 / B0385
Natural Language Processing

Lesson 20. Beyond

Alberto Barrón-Cedeño
a.barron@unibo.it

16/12/2024

Table of Contents

Transformers[1]

[1] Partially based on https://neptune.ai/blog/bert-and-the-transformer-architecture-reshaping-the-ai-landscape

Attention (Vaswani et al., 2017)

- RNNs are [were] at the core of NLU tasks —language modeling, machine translation and question answering

[2] I just passed by a paper with title "pre-training without attention"...

Attention (Vaswani et al., 2017)

- RNNs are [were] at the core of NLU tasks —language modeling, machine translation and question answering
- Attention is all you need[2] introduced the "self-attention" mechanism for MT: en–de and en–fr

[2] I just passed by a paper with title "pre-training without attention"...

Attention (Vaswani et al., 2017)

- RNNs are [were] at the core of NLU tasks —language modeling, machine translation and question answering
- Attention is all you need[2] introduced the "self-attention" mechanism for MT: en–de and en–fr
- Comparison against recurrent and convolutional models:
  - Higher translation quality
  - Less computation cost

[2] I just passed by a paper with title "pre-training without attention"...

Attention (Vaswani et al., 2017)

- RNNs are [were] at the core of NLU tasks —language modeling, machine translation and question answering
- Attention is all you need[2] introduced the "self-attention" mechanism for MT: en–de and en–fr
- Comparison against recurrent and convolutional models:
  - Higher translation quality
  - Less computation cost
- By reading one word at a time, RNNs have a hard time modelling distant word interactions

[2] I just passed by a paper with title "pre-training without attention"...

Attention (Vaswani et al., 2017)

- RNNs are [were] at the core of NLU tasks —language modeling, machine translation and question answering
- Attention is all you need[2] introduced the “self-attention” mechanism for MT: en–de and en–fr
- Comparison against recurrent and convolutional models:
  - Higher translation quality
  - Less computation cost
- By reading one word at a time, RNNs have a hard time modelling distant word interactions
- CNN’s get all the info at once, but combining distant relationships comes late

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

[2] I just passed by a paper with title “pre-training without attention”...

Transformer (Devlin et al., 2019)

- A small/constant number of steps (chosen empirically)

[3] In parallel for all words, multiple times

Transformer (Devlin et al., 2019)

- A small/constant number of steps (chosen empirically)
- The self-attention mechanism models relationships between all words in a sentence, regardless of their respective position

[3] In parallel for all words, multiple times

Transformer (Devlin et al., 2019)

- A small/constant number of steps (chosen empirically)
- The self-attention mechanism models relationships between all words in a sentence, regardless of their respective position
- Attention: scores that determine how much each of the other words should contribute to the next representation of each of them

[3] In parallel for all words, multiple times

Transformer (Devlin et al., 2019)

- A small/constant number of steps (chosen empirically)
- The self-attention mechanism models relationships between all words in a sentence, regardless of their respective position
- Attention: scores that determine how much each of the other words should contribute to the next representation of each of them

Example:

I arrived at the bank after crossing the river
I arrive at the bank after crossing the road

[3] In parallel for all words, multiple times

Transformer (Devlin et al., 2019)

- A small/constant number of steps (chosen empirically)
- The self-attention mechanism models relationships between all words in a sentence, regardless of their respective position
- Attention: scores that determine how much each of the other words should contribute to the next representation of each of them

Example:

I arrived at the bank after crossing the river
I arrive at the bank after crossing the road

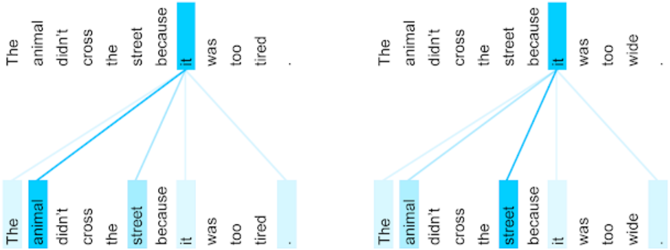
Let us look at an animated example for MT: transform20fps.gif

1. Initial embedding representations (empty circles)
2. new representation (filled circles) ← aggregating info (attention) from all other words (context)[3]

[3] In parallel for all words, multiple times

Transformer (Devlin et al., 2019)

The attention can be *observed*, here within two contexts:

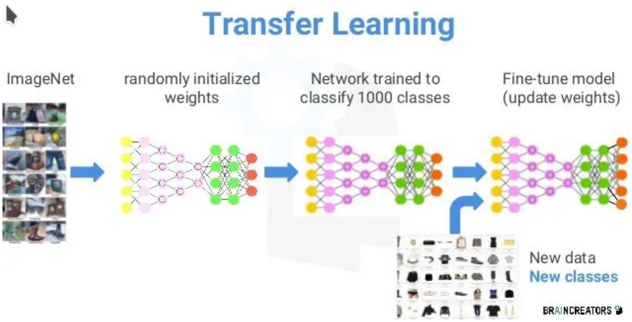


How to translate it in these cases?

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Pre-trained models

Transfer learning (image recognition, again)

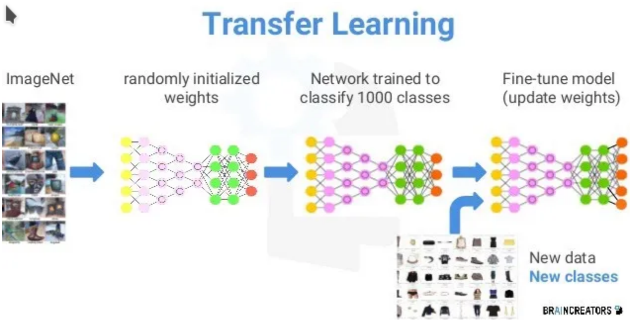


1. Train a model on a (large) [open,out-of]-domain corpus
2. Fine-tune it with new data to your task of interest

Picture from `https://madhuramiah.medium.com/deep-learning-using-resnets-for-transfer-learning-d7f4799fa863`

Pre-trained models

Transfer learning (image recognition, again)



1. Train a model on a (large) [open,out-of]-domain corpus
2. Fine-tune it with new data to your task of interest

* Change of paradigm wrt, for instance, word2vec

Picture from `https://madhuramiah.medium.com/deep-learning-using-resnets-for-transfer-learning-d7f4799fa863`

Pre-trained models

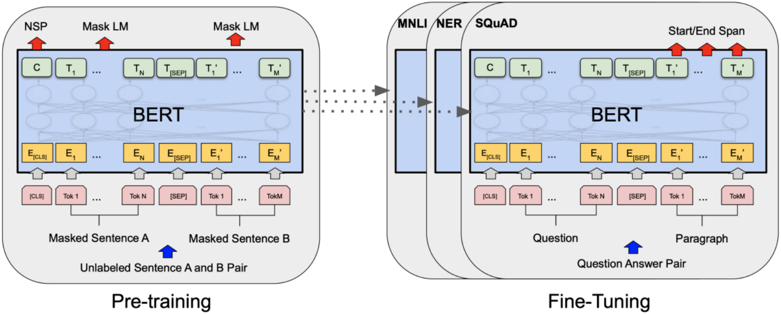
Typical current setting

1. An organisation with large computing capabilities trains a large language model[4]
2. Download and fine-tune the model with a few thousand instances[5]

[4] GPT-3 is trained on 45TB of data; it has 175B parameters

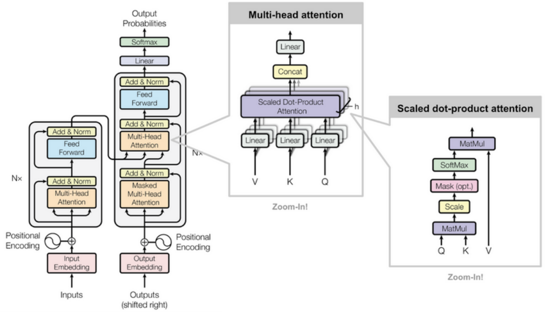
[5] Or even less: zero-shot and few-shot learning; e.g., Muti and Barrón-Cedeño (2022)

Fine-Tuning



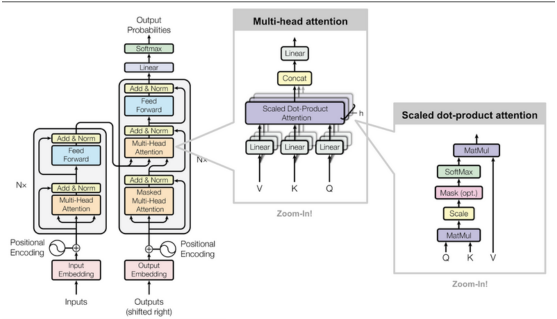
Picture from Devlin et al. (2019)

Transformer architecture[6]



[6] Don't panic!

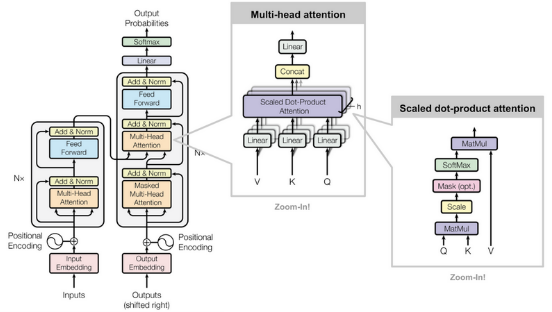
Transformer architecture[6]



- Scaled dot-product attention multiple times, in parallel
- Similar to looping over an RNN, without vanishing gradient descent

[6] Don't panic!

Transformer architecture[6]



- Scaled dot-product attention multiple times, in parallel
- Similar to looping over an RNN, without vanishing gradient descent

Multiple times?

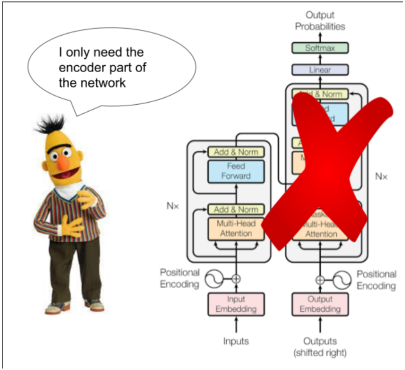
BERT : 24 attention layers
GPT-2 : 12 attention layers
GPT-3 : 96 attention layers

[6] Don't panic!

Bert

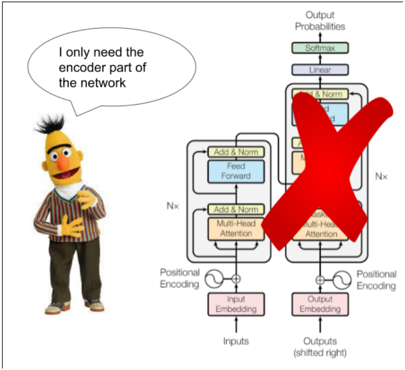
BERT

Bi-directional encoder representations from transformers



BERT

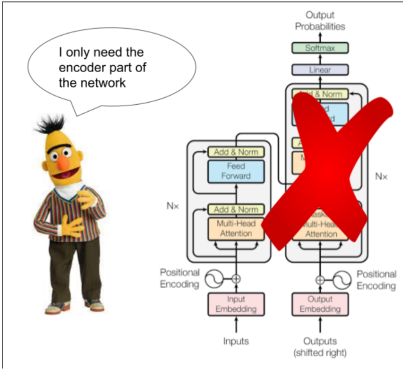
Bi-directional encoder representations from transformers



- Encodes the semantic and syntactic information in the embedding[a]

BERT

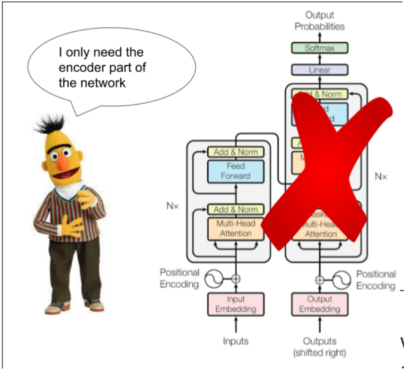
Bi-directional encoder representations from transformers



- Encodes the semantic and syntactic information in the embedding[a]
- No decoding: it's output is an embedding, not text or a class (e.g., to compute similarities; bertscore)

BERT

Bi-directional encoder representations from transformers



- Encodes the semantic and syntactic information in the embedding[a]
- No decoding: it's output is an embedding, not text or a class (e.g., to compute similarities; bertscore)
- Extra training layer: predicts hidden or masked words to force the encoder to learn more about the context

[a] Not for text generation (it can generate words),
allows for multiple languages

BERT
Masking (cloze test)

- When training to predict the next word, BERT might cheat and just copy it from the right-to-left component

BERT

Masking (cloze test)

- When training to predict the next word, BERT might cheat and just copy it from the right-to-left component
- Instead of predicting the next word, we hide or “mask” a word, and then force the model to predict that word

BERT
Masking (cloze test)

- When training to predict the next word, BERT might cheat and just copy it from the right-to-left component
- Instead of predicting the next word, we hide or "mask" a word, and then force the model to predict that word
  - 15% of the input tokens are masked (picked randomly):

%	masked with	Sentence
	(original)	BERT can see all the words in this sentence

BERT

Masking (cloze test)

- When training to predict the next word, BERT might cheat and just copy it from the right-to-left component
- Instead of predicting the next word, we hide or “mask” a word, and then force the model to predict that word
  - 15% of the input tokens are masked (picked randomly):

%	masked with	Sentence
	(original)	BERT can see all the words in this sentence
80	MASK token	BERT can see all the [MASK] in this sentence

BERT

Masking (cloze test)

- When training to predict the next word, BERT might cheat and just copy it from the right-to-left component
- Instead of predicting the next word, we hide or "mask" a word, and then force the model to predict that word
  - 15% of the input tokens are masked (picked randomly):

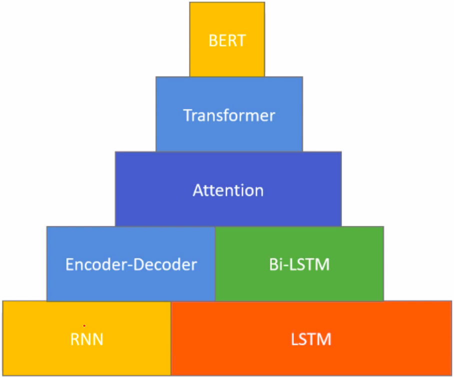
%	**masked with**	**Sentence**
	(original)	BERT can see all the words in this sentence
80	MASK token	BERT can see all the [MASK] in this sentence
10	random word	BERT can see all the ragù in this sentence

- When training to predict the next word, BERT might cheat and just copy it from the right-to-left component
- Instead of predicting the next word, we hide or "mask" a word, and then force the model to predict that word
  - 15% of the input tokens are masked (picked randomly):

%	**masked with**	**Sentence**
	(original)	BERT can see all the words in this sentence
80	MASK token	BERT can see all the [MASK] in this sentence
10	random word	BERT can see all the ragù in this sentence
10	same word	BERT can see all the words in this sentence

BERT

Learning Pyramid



Picture from https://iq.opengenus.org/introduction-to-bert/

BERT in other Languages

For instance:

- Spanish (Cañete et al., 2020)
- Italian (AlBERTo) (Polignano et al., 2019)

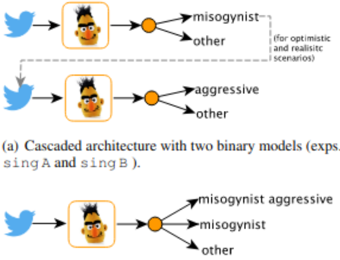
(Muti and Barrón-Cedeño, 2020)

BERT in other Languages

For instance:

- Spanish (Cañete et al., 2020)
- Italian (AlBERTo) (Polignano et al., 2019)

Use case: misogyny identification in Italian



(a) Cascaded architecture with two binary models (exps. `singA` and `singB`).

(b) Multi-class architecture model (exp. `multi`).

Figure 1: The two alternative system architectures for misogyny and aggressiveness identification.

(Muti and Barrón-Cedeño, 2020)

Multilingual models

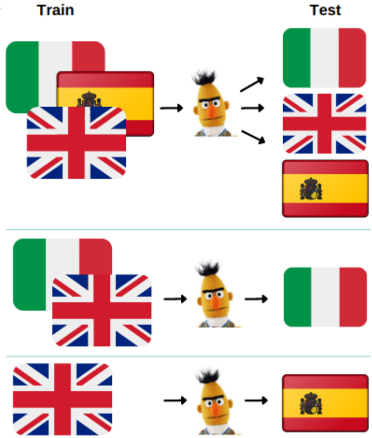
What makes multilingual BERT multilingual? (Liu et al., 2020)

(Muti and Barrón-Cedeño, 2022)

Multilingual models

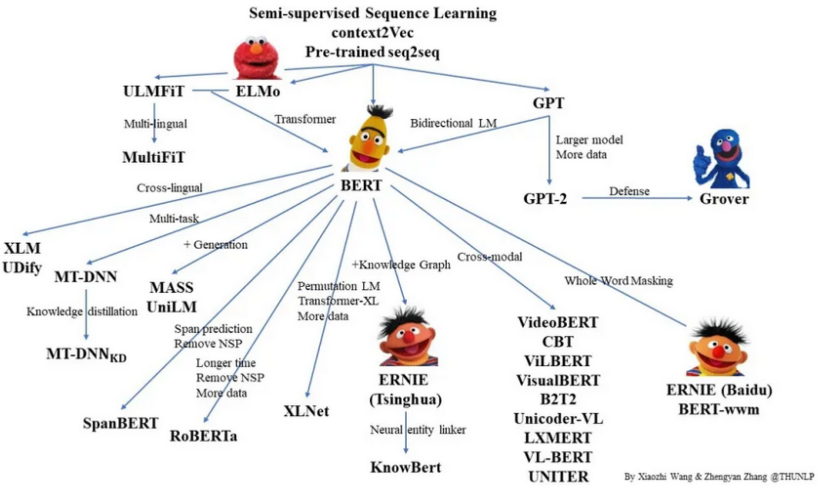
What makes multilingual BERT multilingual? (Liu et al., 2020)

Use case: multilingual misogyny identification



(Muti and Barrón-Cedeño, 2022)

BERTology



Picture from https://github.com/thunlp/

(Other) Reference Libraries

- Spacy
  Industrial-Strength Natural Language Processing
  https://spacy.io/
- Stanza
  A Python NLP Package for Many Human Languages
  https://stanfordnlp.github.io/stanza/
- Hugging Face
  The AI community building the future
  https://huggingface.co/

Conferences (non-exhaustive)

NLP-ish	IR-ish	MT-ish
Top		
ACL	SIGIR	WMT
EMNLP	CIKM	EAMT
NAACL	WSDOM	
EACL	ECIR	
Nice		
SemEval	CLEF	
CICLing[7]	TREC	
LREC		
National		
CLIC-it	IIR	
Evalita		

[7] Apparently gone

Recap

Recap: The path

1. Baby steps into computing

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording. . .

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks
10. From words to semantics: word embeddings

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks
10. From words to semantics: word embeddings
11. Taking snapshots of text: CNNs

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks
10. From words to semantics: word embeddings
11. Taking snapshots of text: CNNs
12. Texts as sequences: (Bi)RNNs

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks
10. From words to semantics: word embeddings
11. Taking snapshots of text: CNNs
12. Texts as sequences: (Bi)RNNs
13. Using a better memory: LSTM

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks
10. From words to semantics: word embeddings
11. Taking snapshots of text: CNNs
12. Texts as sequences: (Bi)RNNs
13. Using a better memory: LSTM
14. LSTM to produce text

Recap: The path

1. Baby steps into computing
2. What is NLP? From rule-based to statistical
3. Pre-processing text: tokens, stemming, stopwording...
4. From words to vectors: the vector space model
5. A few supervised models
6. Training and evaluating in machine learning
7. From words to meaning: topic modeling
8. Using one *neuron*: perceptron
9. Fully-connected neural networks
10. From words to semantics: word embeddings
11. Taking snapshots of text: CNNs
12. Texts as sequences: (Bi)RNNs
13. Using a better memory: LSTM
14. LSTM to produce text
15. Intro to transformers

Recap: The future path

- We covered Parts 1 and 2 of Lane et al. (2019) (up to Section 9)
- That's 9 out of 13 chapters of Natural Language Processing in Action

Now go and celebrate the end of the course

...and worry about your project from Jan 2nd!

- I'm available during January for 1-to-1 discussion on your project upon request!

References I

Cañete, J., G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez
2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova
2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Lane, H., C. Howard, and H. Hapkem
2019. *Natural Language Processing in Action*. Shelter Island, NY: Manning Publication Co.

Liu, C.-L., T.-Y. Hsu, Y.-S. Chuang, and H. yi Lee
2020. What makes multilingual bert multilingual? *arXiv*.

Muti, A. and A. Barrón-Cedeño
2020. UniBO@AMI: A Multi-Class Approach to Misogyny and Aggressiveness Identification on Twitter Posts Using AlBERTo. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*.

References II

Muti, A. and A. Barrón-Cedeño
2022. A checkpoint on multilingual misogyny identification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Pp. 454–460, Dublin, Ireland. Association for Computational Linguistics.

Polignano, M., P. Basile, M. de Gemmis, G. Semeraro, and V. Basile
2019. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, volume 2481, Bari, Italy. CEUR.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin
2017. Attention is all you need.