BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

Multimodal Dialogue Systems Seminar
Abdallah Bashir
CS UdS

19 Sep 2019
Saarbrucken
Germany

Outline

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

Keywords!

- Transformers
- Contextual Embeddings
- Bidirectional Training
- Fine-Tuning
- SOTA, alot of SOTA!

Outline

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

Introduction

- 2018 was a turning point for Natural Language Processing
    - BERT
    - OpenAI GPT
    - ELMO
    - ULM fit

[1]

Introduction

- BERT [2] (Bidirectional Encoder Representations from Transformers) is an Open-Source Language Representation Model developed by researchers in Google AI.
- At that time, the paper presented SOTA results in eleven NLP tasks.

Models that outperformed bert mentioned at the end.

Main Contributions

The paper summarizes the contributions in main three points

- demonstrate the importance of Bidirectional pre-training and why it is better than unidirectional approach.
- With the usage of fine-tuning, BERT outperforms task-specific architectures.
- Showing the performance achievements where BERT advances the SOTA for eleven NLP tasks

Outline

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

Related Work

- The paper briefly go through previous approaches in pre-training general language representations, which are listed under main three points:
    - Unsupervised Feature-based Approaches
    - Unsupervised Fine-tuning Approaches
    - Transfer Learning from Supervised Data

Unsupervised Feature-based Approaches

- Approaches to Learning representations from unlabeled text, i.e word embeddings included **non-neural approaches** and **neural approaches**.
- Using pre-trained word-embeddings instead of training it from scratch have proved significant improvements in performance.

Non Contextual Embeddings

-0.34	-0.84	0.20	-0.26	-0.12	0.23	1.04	-0.16	0.31	0.06	0.30	0.33	-1.17	-0.30	0.03	0.09	0.35	-0.28	-0

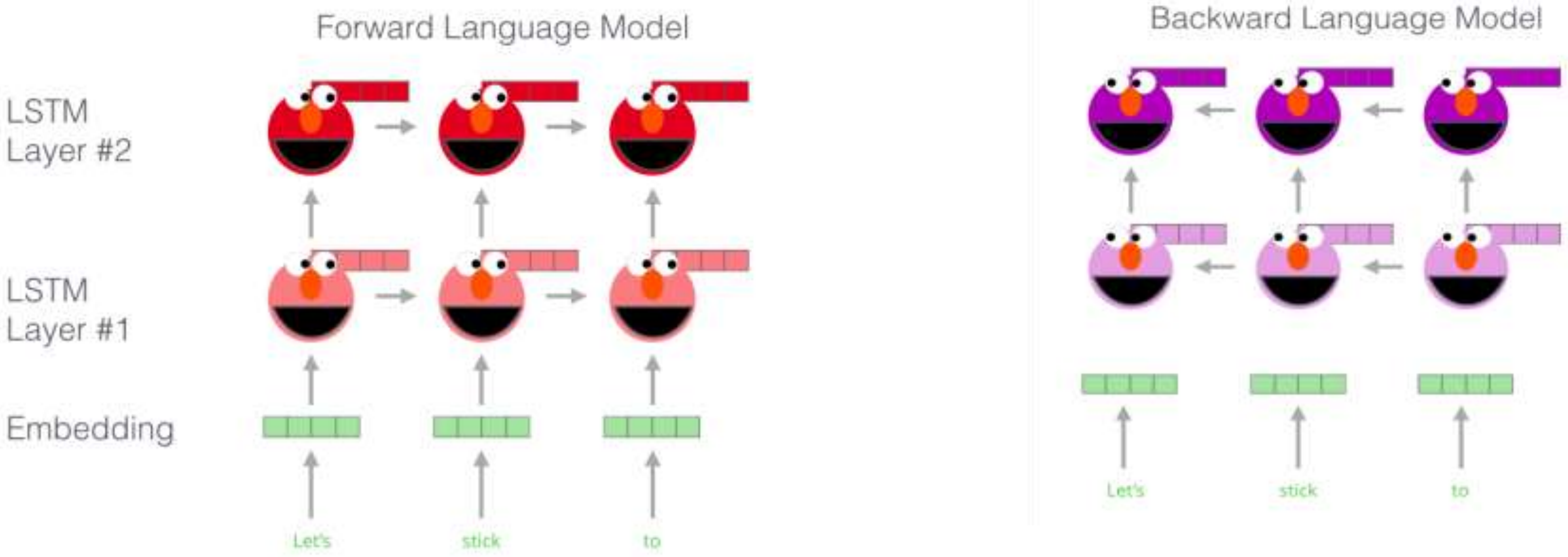
The GloVe word embedding of the word "stick" - a vector of 200 floats (rounded to two decimals). It goes on for two hundred values. [1]

Contextual Embeddings

- Unidirectional
    - Feature-Based(ELMo)
    - Fine-tuning(OpenAI GPT).
- Bidirectional
    - BERT

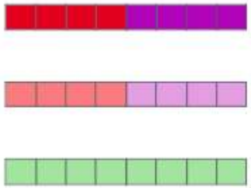
ELMO, What about sentences' context

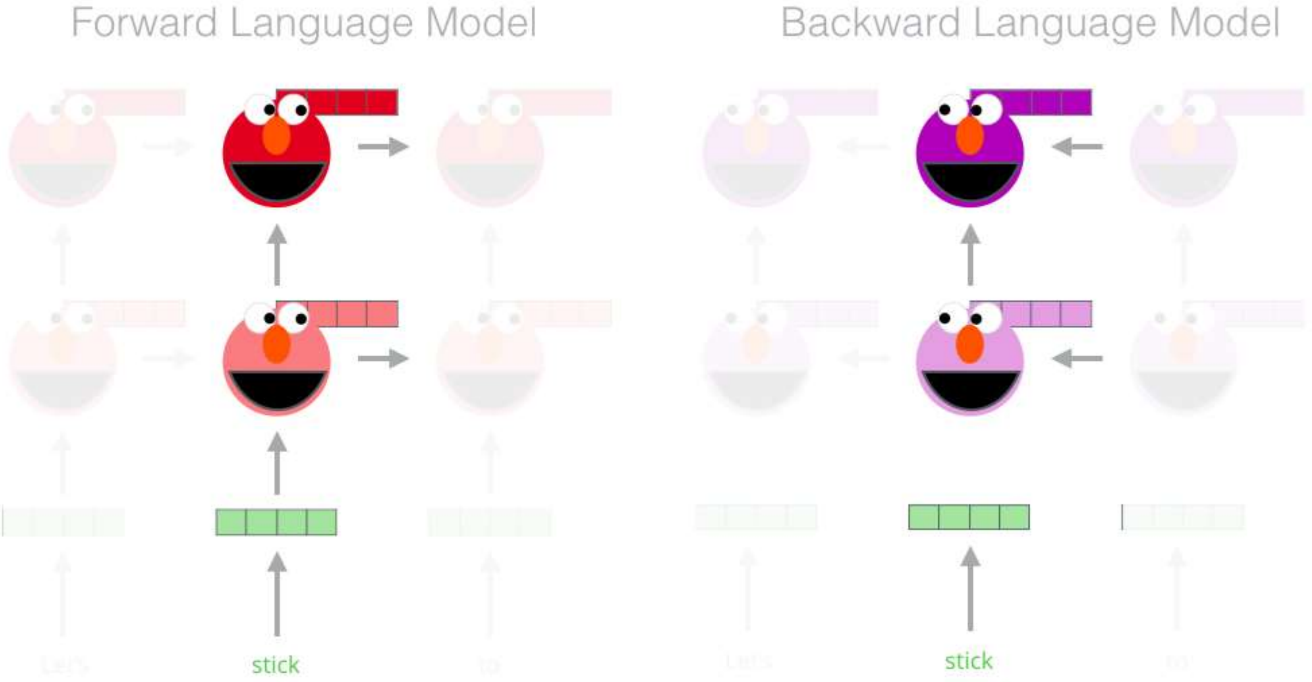
Embedding of “stick” in “Let’s stick to” - Step #1



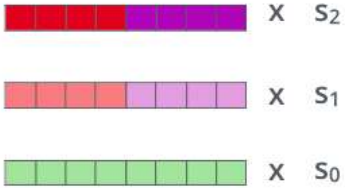
Embedding of “stick” in “Let’s stick to” - Step #2

1- Concatenate hidden layers





2- Multiply each vector by a weight based on the task



3- Sum the (now weighted) vectors

ELMo embedding of “stick” for this task in this context

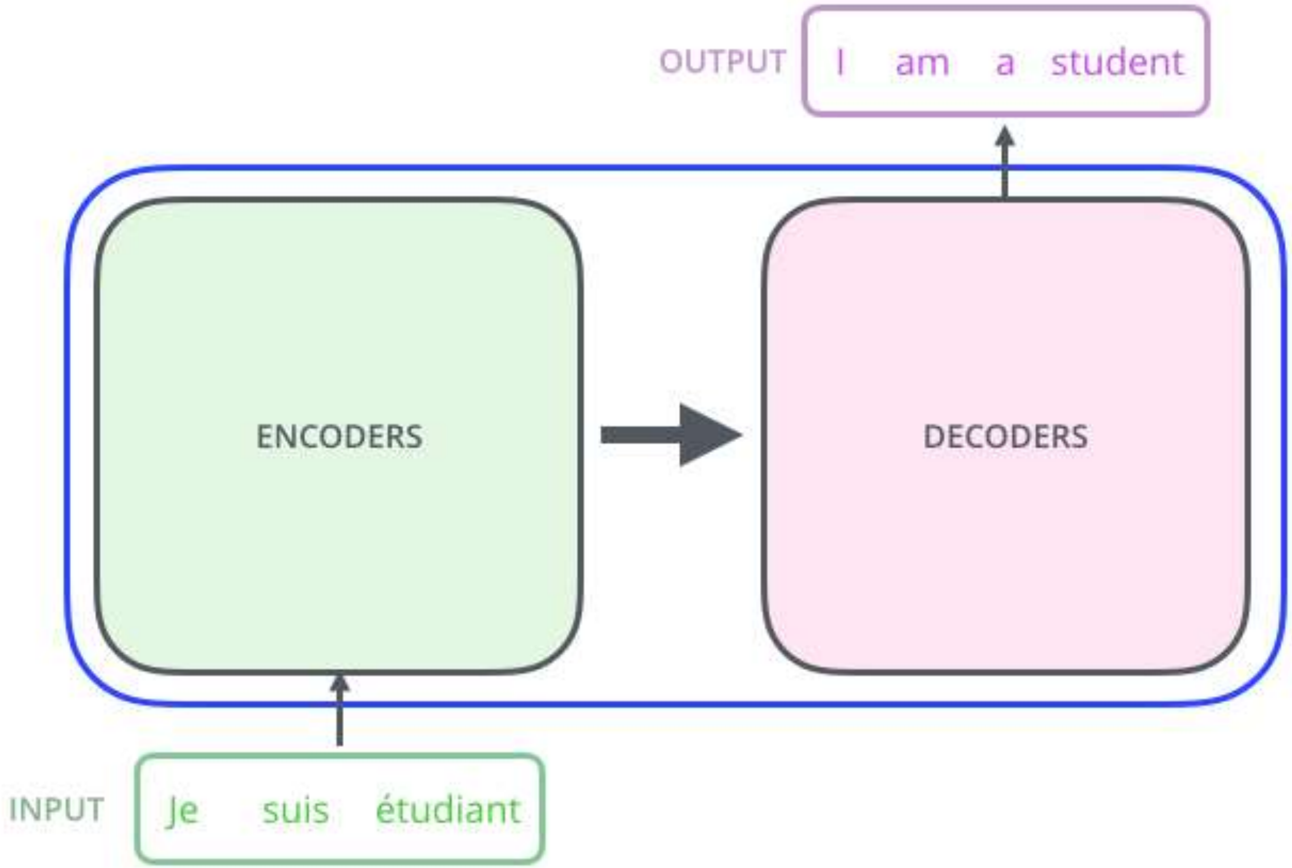
[16]

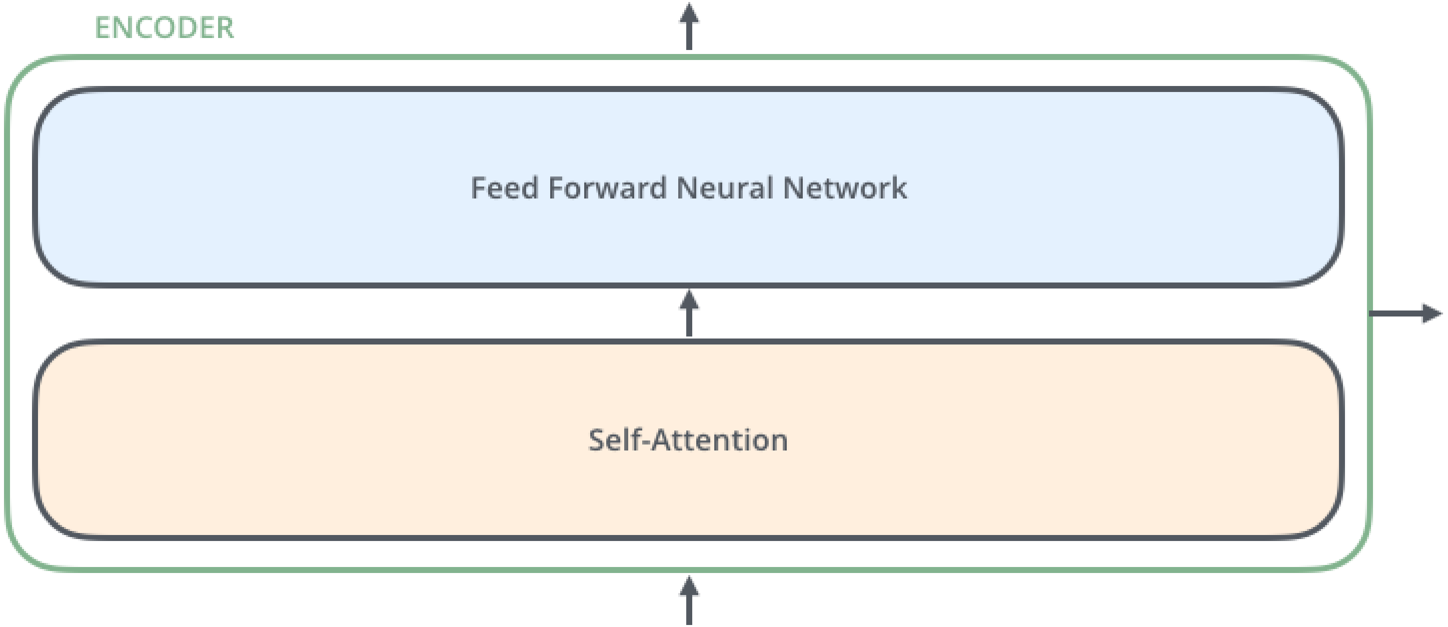
- When ELMo contextual representations was used in task-specific architecture, ELMo advanced the SOTA benchmarks.

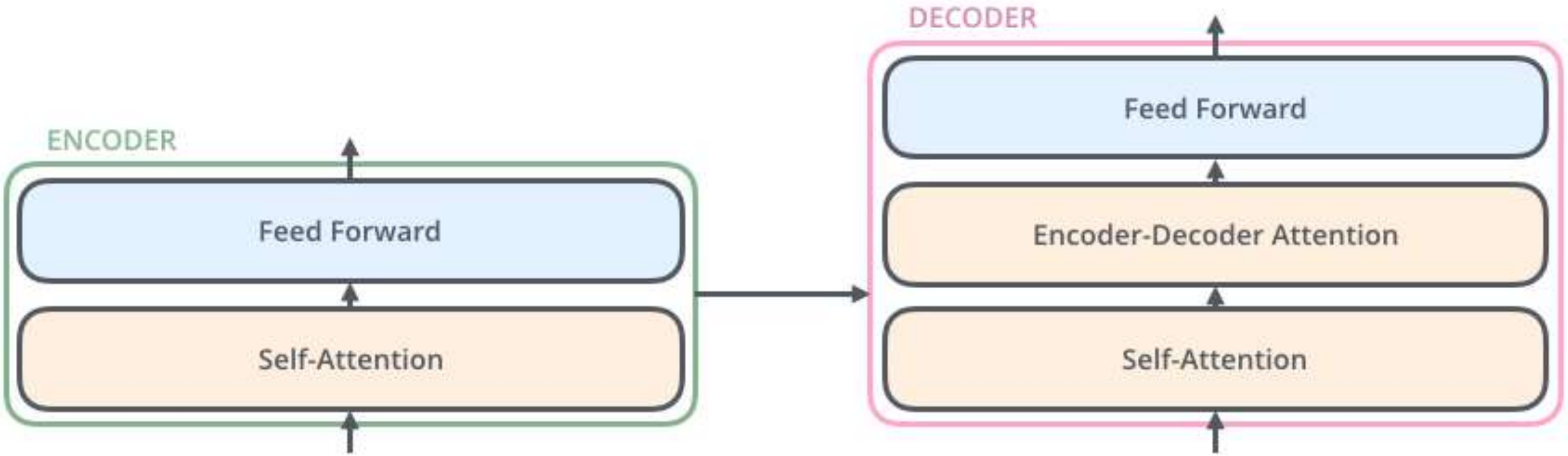
Unsupervised Fine-tuning Approaches

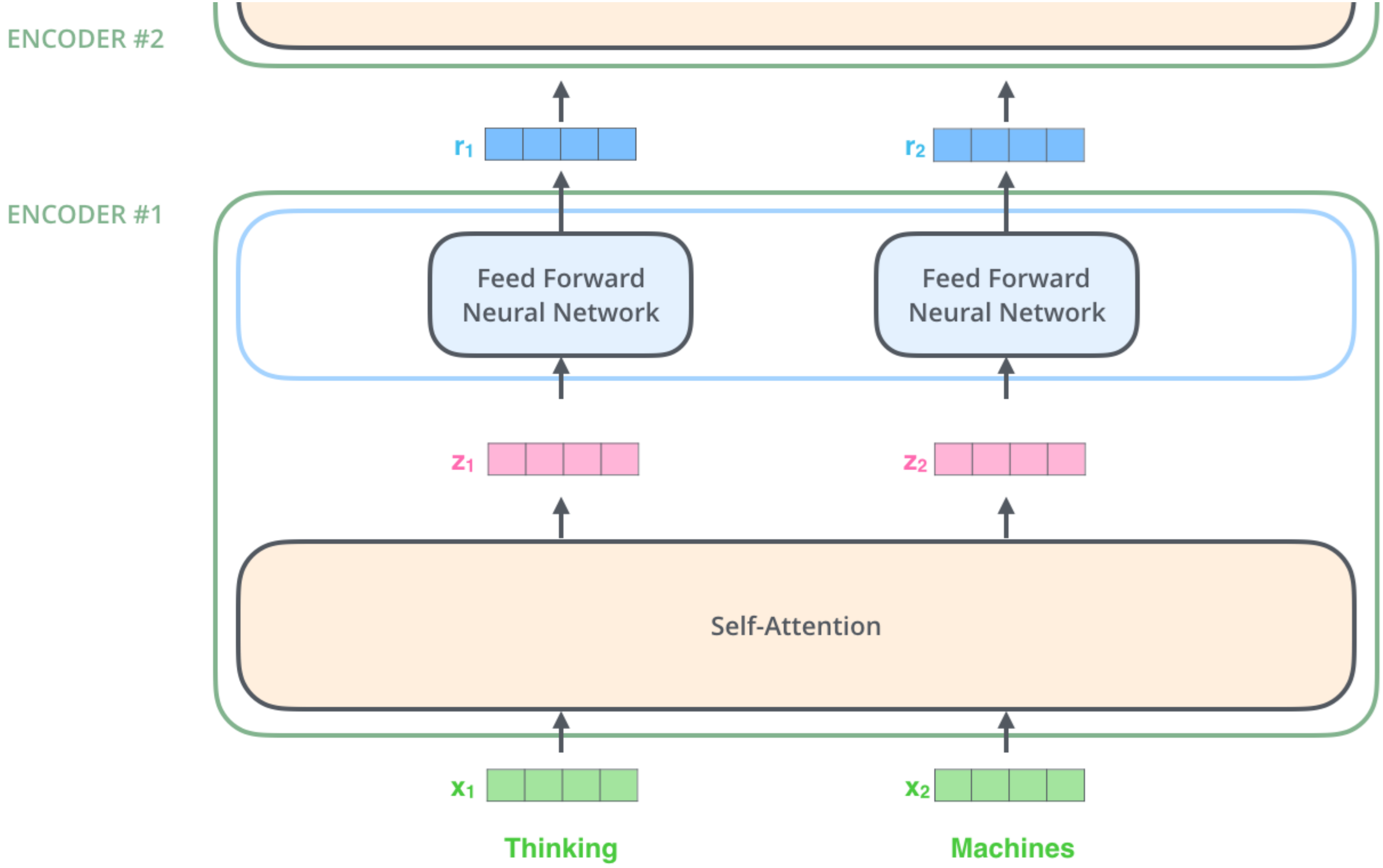
Transformer


INPUT
Je suis étudiant
THE TRANSFORMER
OUTPUT
I am a student

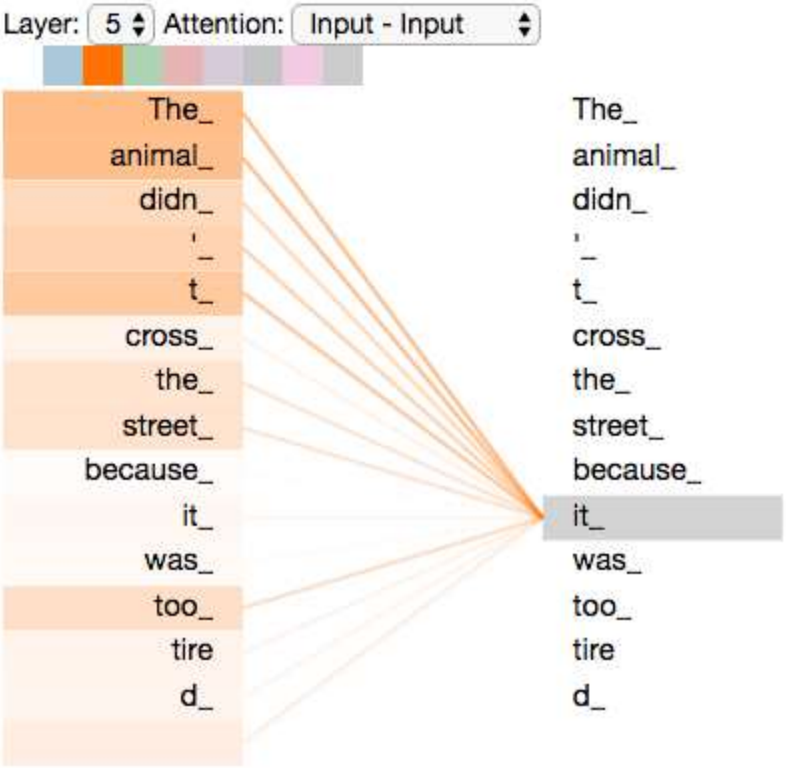

OUTPUT
I am a student
ENCODERS
DECODERS
INPUT
Je suis étudiant


ENCODER
Feed Forward Neural Network
Self-Attention


ENCODER
Feed Forward
Self-Attention
DECODER
Feed Forward
Encoder-Decoder Attention
Self-Attention


ENCODER #2
r1
r2
ENCODER #1
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
Self-Attention
x1
x2
Thinking
Machines

Self-Attention in a Nutshell!

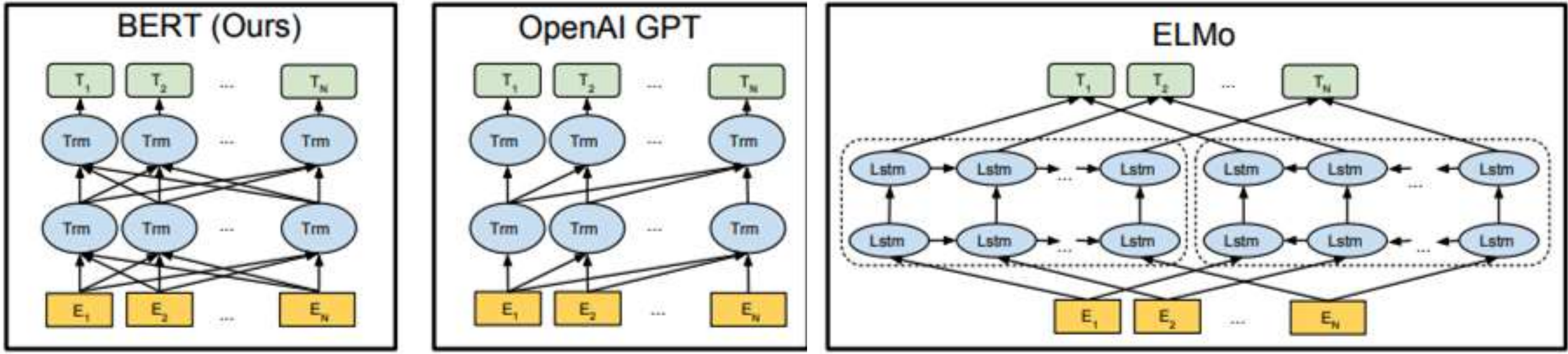


Unsupervised Fine-tuning Approaches

- Perks of using such approaches that only few parameters need to be trained from scratch.
- OpenAI GPT [4] previously achieved SOTA on many sentence level tasks in GLUE benchmark [19]

Transfer Learning from Supervised Data

- Transfer learning is a means to extract knowledge from a source setting and apply it to a different target setting.
- Researchers in the field of Computer Vision have used transfer learning continuously where they fine-tune models pre-trained with ImageNet.


BERT (Ours)
OpenAI GPT
ELMo
T1 T2 ... TN
Trm
Lstm
E1 E2 ... EN

Outlines

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

BERT | The Model

- BERT implementation:
  - **Pretraining**
  - **Fine-tuning**

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step

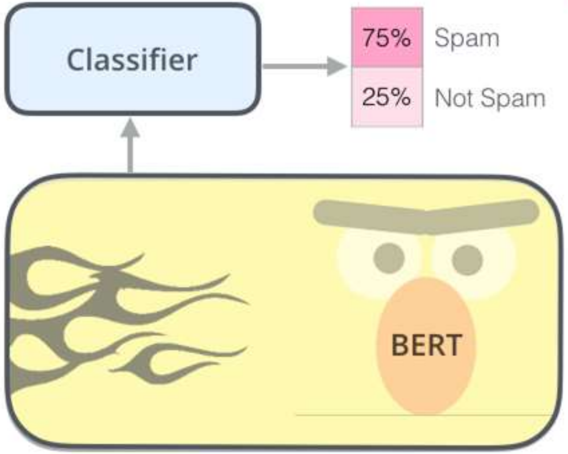
Model:

Dataset:

Objective: Predict the masked word (langauge modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step



Model: (pre-trained in step #1)

Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

5]

- BERT almost have the same model architecture across different tasks with small changes between the pre-trained architecture and the final downstream architecture

Model Architecture

- BERT architecture consist of multi-layer bidirectional Transformer encoder [5].
- BERT model provided in the paper came in two sizes

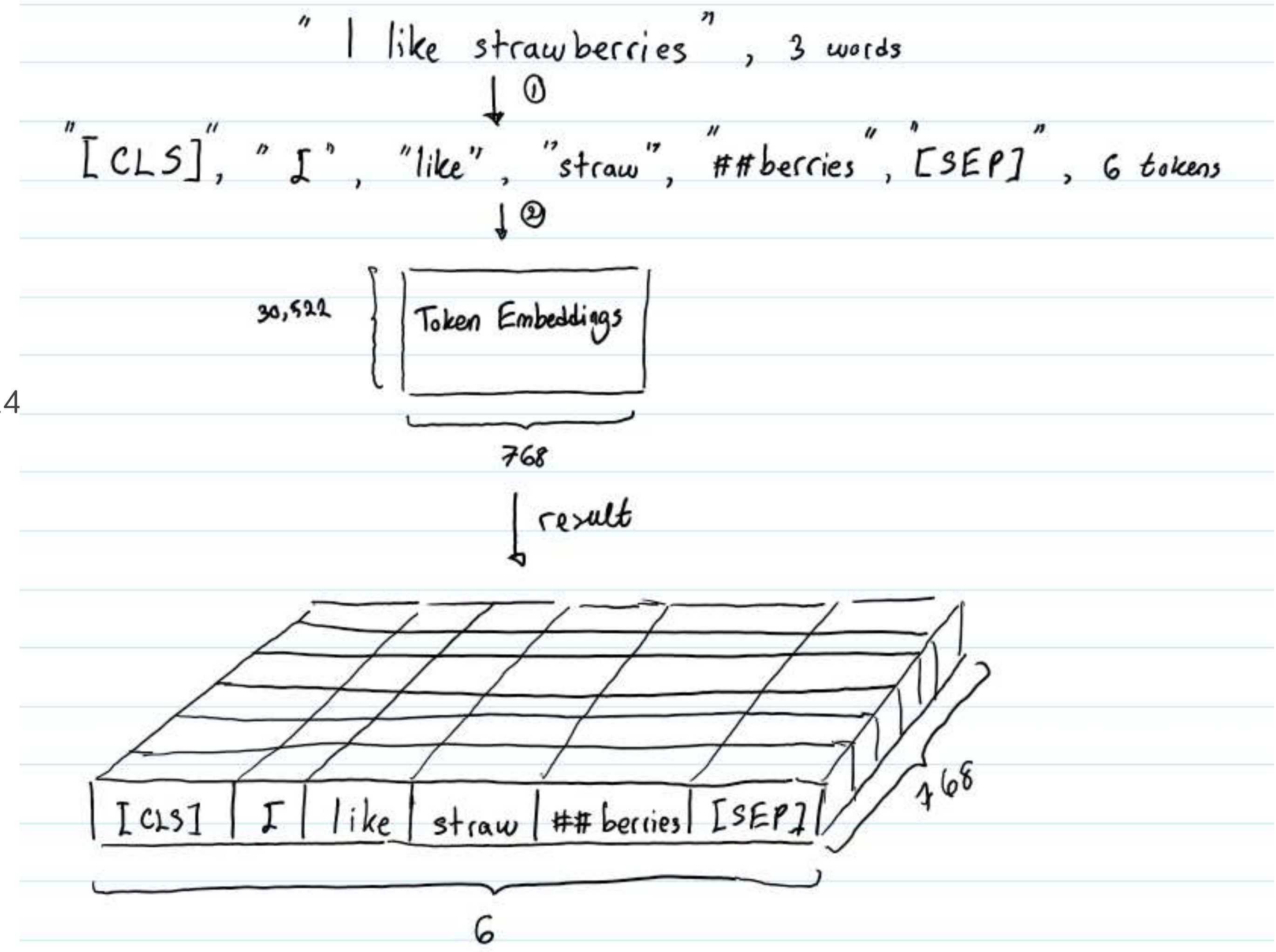
$BERT_{BASE}$	$BERT_{LARGE}$
Layers = 12	Layers = 24
Hidden size = 768	Hidden size = 1024
self-Attention heads = 12	self-Attention heads = 16
Total parameters = 110M	Total parameters = 340M

Comparisons with OpenAI GPT

- $\text{BERT}_{\text{base}}$ was chosen to have the same size as OpenAI GPT for benchmarks purposes.
- The difference between the two models is BERT train the Language Model transformers in both directions while the OpenAI GPT context trains it only left-to-right.

Model Input/Output Representation

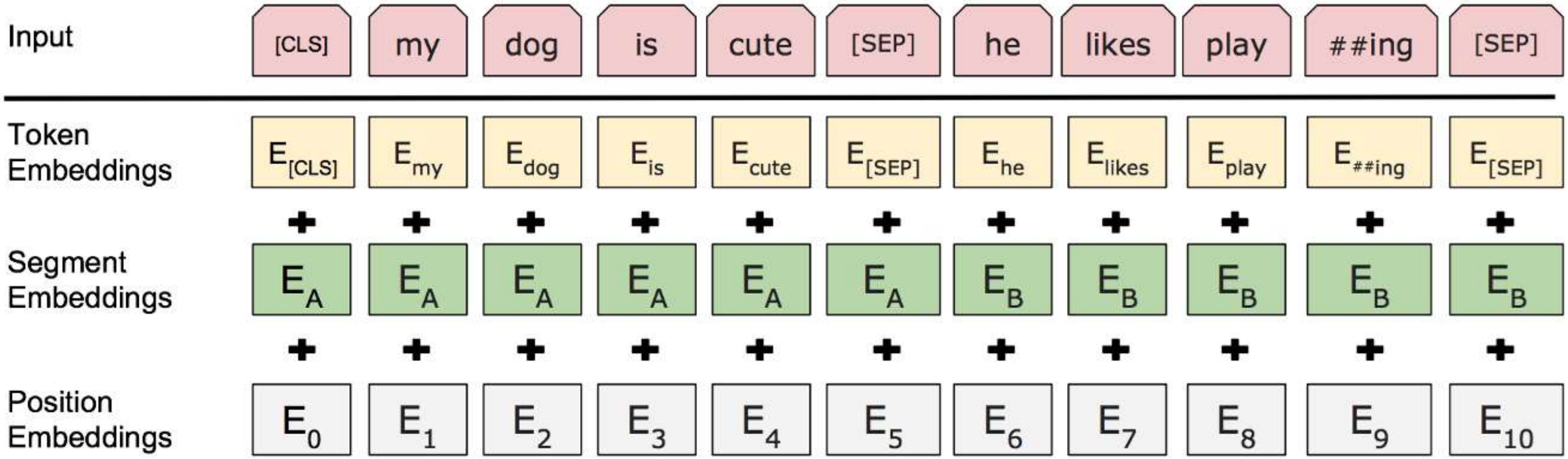
- The paper defines a ‘sentence’ as any sequence of text. Like mentioned before, it could be one sentence or two sentences stacked together.
- BERT input can be represented by either a single sentence, or pair of sentences (like Question Answering).
- BERT use WordPiece embeddings [23] with 30,000 token vocabulary.


" I like strawberries " , 3 words
①
"[CLS]", " I ", "like", "straw", "##berries", "[SEP]" , 6 tokens
②
30,522
Token Embeddings
768
result
[CLS]
I
like
straw
##berries
[SEP]
768
6

[24

Model Input/Output Representation

- The first token in each sequence is always a [CLS] token that is used in classification.
- Separating Sentences apart can be done in two steps:
  - Separate between sentences using [SEP] token,
  - Adding a learned embedding to each token in order to state which sentence it belongs to.



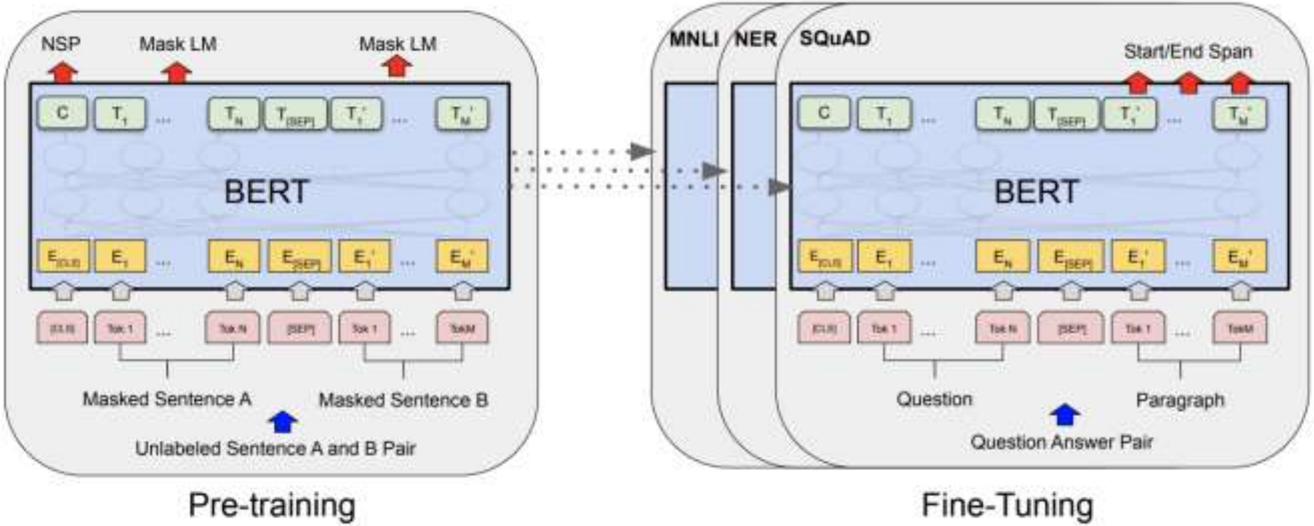
- Each token is represented by summing the corresponding token, segment and position embedding as seen in the above figure

Outline

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

Pre-training BERT

- In order to pre-train BERT transformers bidirectionally the paper proposed two unsupervised tasks which are **Masked LM** and **Next Sentence Prediction(NSP)**.



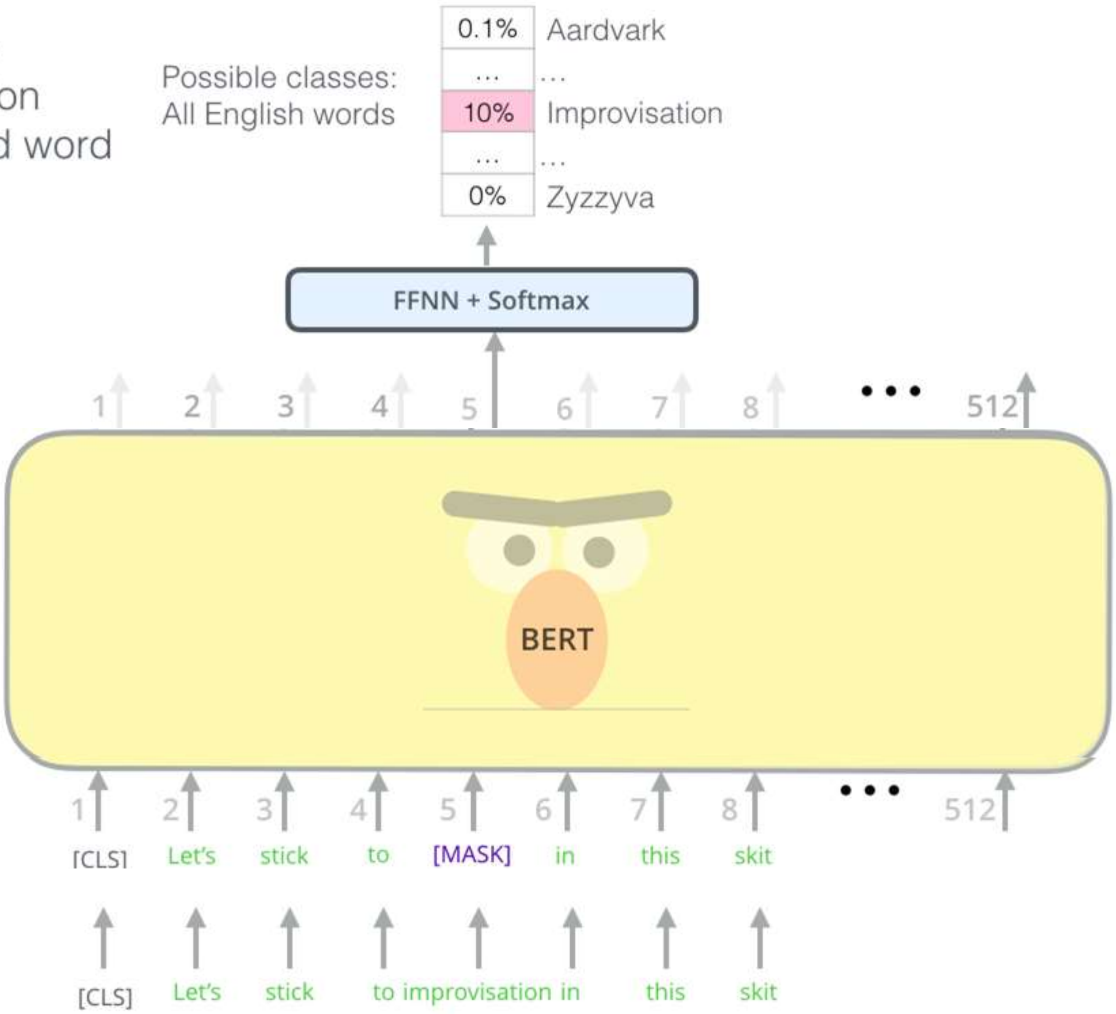
Pre-training Data

- One of the reasons of BERT success is the amount of data that it got trained on.
- The main two corpuses that were used to pretrain the language models are:
  - the BooksCorpus (800M words)
  - English Wikipedia (2,500M words)

Masked Language Modeling

- Usually, Language Models can only be trained on one specific direction.
- BERT handled this issue by using a "Masked Language Model" from previous literature (Cloze [24]).
- BERT do this by randomly masking **15%** of the wordpiece input tokens. Only the masked tokens get to be predicted later.

Use the output of the masked word's position to predict the masked word



Randomly mask 15% of tokens

Input

Masked Language Modeling

- [MASK] tokens would appear only in the pretraining and not during the fine-tuning!
- To alleviate the masking, after choosing the randomly 15% tokens, BERT either:
  - Switch the token to [MASK] (80% of the time).
  - Switch the token to a random other token (10% of the time).
  - Leave the token unchanged (10% of the time).
- The original token is then predicted with cross entropy loss.

Next Sentence Prediction (NSP)

- In the pretraining phase, the model (if provided) receives pairs of sentences and it will be trained to predict if the second sentence is the subsequent of the first.
- In the training data, **50%** of the inputs are actual pairs with a label *[IsNext]*, where the other 50% have random sentences from the corpus as the successor for the first sentence [NotNext].
- **NSP** is crucial for downstream tasks like Question Answering and Natural Language Inference.

Next Sentence Prediction (NSP)

Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

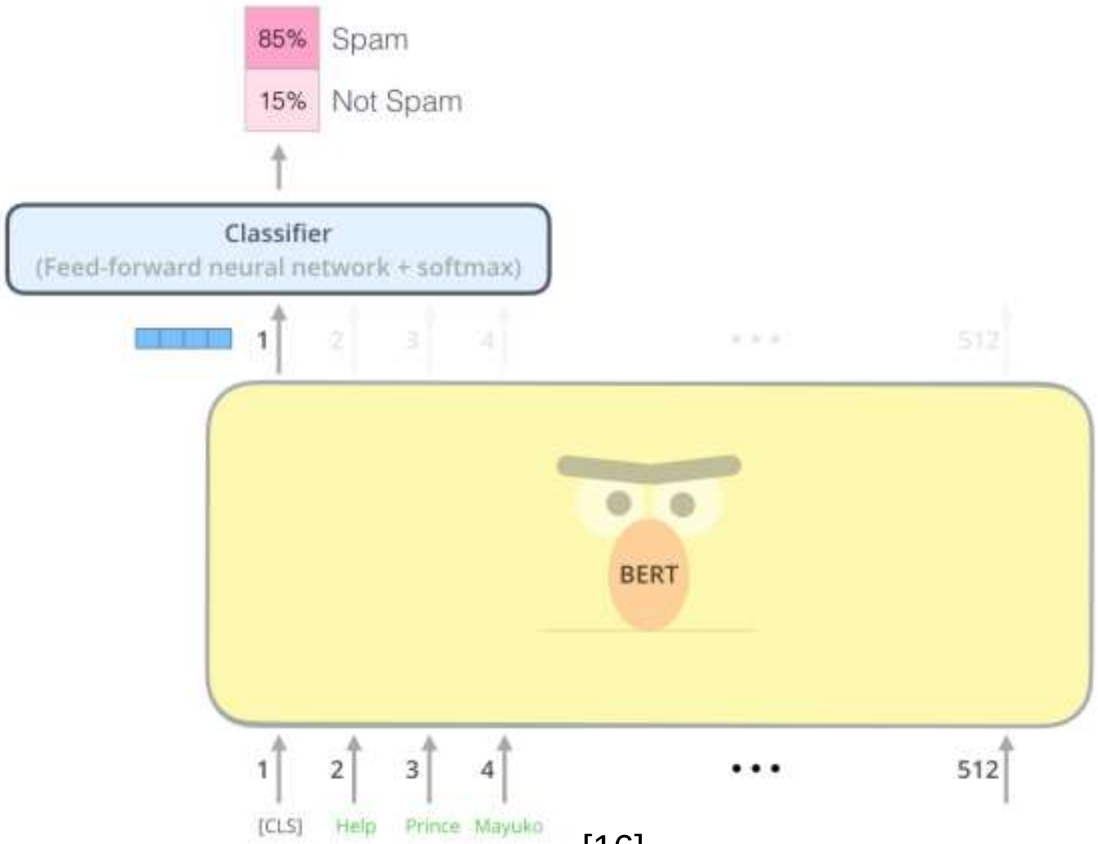
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

Fine-tuning BERT

- this stage in total is considered to be inexpensive relative to the pretraining phase.
- For **classification tasks** (e.g sentiment analysis) we add a classification FFN for the [CLS] token input representation on top of the final output.
- For **Question Answering** alike tasks Bert train two extra vectors that are responsible for marking the beginning and the end of the answer.

Classification Example



[16]

Outline

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

Experiments


SENORGIF.COM
YES

Experiments

- GLUE
  - General Language Understanding Evaluation benchmark
- SQuAD v1.1
  - Stanford Question Answering Dataset
- SQuAD v2.0
  - extends the SQuAD 1.1
- SWAG
  - Situations With Adversarial Generations (SWAG) dataset

1. GLUE

- “The General Language Understanding Evaluation benchmark (GLUE) [19] is a collection of various natural language understanding tasks.”
- For fine-tuning,the same approach of classification in pre-training is used.

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results [2]

2. SQuAD v1.1

- "SQuAD v1.1 [26], The Stanford Question Answering Dataset is a collection of 100k crowdsourced question,answer pairs."
- Input get represented as one sequence containing the question and the text containing the answer.

2. SQuAD v1.1

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results

3. SQuAD v2.0

- The SQuAD 2.0 task extends the SQuAD 1.1 by:
  - making sure that no short answer exists in the provided paragraph.
  - marking questions that do not have an answer with [CLS] token at the start and the end. The TriviaQA data was used for this model.

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
BERT$_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results

4. SWAG

- “The Situations With Adversarial Generations (SWAG) dataset contains 113k sentence-pair completion examples that evaluate grounded common sense inference [28],
- Given a sentence, the task is to choose the most plausible continuation among four choices”
- The paper fine-tune on the SWAG dataset by creating four input sequences, each include the given sentence and concatenated with a possible continuation.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Resultس

Outline

- Introduction
- Related Works
- BERT | The Model
- Pre-training BERT
- Experiments
- Ablation Studies

Ablation Studies

1. Effect of Pre-training Tasks

- The next experiments will showcase the importance of the bidirectionality of BERT.
- the same pretraining data, fine-tuning scheme, and hyperparameters as $BERT_{BASE}$ will be used throughout the experiments.

1. Effect of Pre-training Tasks

Experiments	Description
No NSP	bidirectional model trained only using the Masked Language Model (**MLM**) without the Next Sentence Prediction **NSP.**
LTR & No NSP	A standard Left-to-Right (LTR) language model which is trained without the **MLM** and the **NSP.** this model can be compared to the OpenAI GPT.

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$BERT_{BASE}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the $BERT_{BASE}$ architecture [2]

2. Effect of Model Size

- The effect of Bert model size on fine-tuning tasks was tested with different number of layers, hidden units, and attention heads while using the same hyperparameters.
- Results from fine-tuning on GLUE are shown in Table 6 which include the average Dev Set accuracy.
- It is clear that the larger the model, the better the accuracy.

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

Table 6: Ablation over BERT model size [2]

- Bert shows for the first time that increasing the model can improve the results even for downstream tasks that its data is very small because it benefit from the larger, more expressive pre-trained representations.

3. Feature-based Approach with BERT

- There are other ways to use Bert for downstream tasks other than fine-tuning which is using the contextualized word embeddings that are generated from pre-training BERT, and then use these fixed features in other models.
- Advantages of using such an approach:
  - Some tasks require a task-specific model architecture and can't be modeled with a Transformer encoder architecture.
  - Computational efficient

3. Feature-based Approach with BERT

- The two approaches were compared by applying Bert to the CoNLL-2003 Named Entity Recognition (NER) task.
- These contextual embeddings are used as input to a randomly initialized two-layer 768-dimensional BiLSTM before the classification layer.
- The paper tried different approaches to represent the embeddings (all shown in Table 7.) in the feature-based approach. Concatenate the last four-hidden layers output achieved the best.

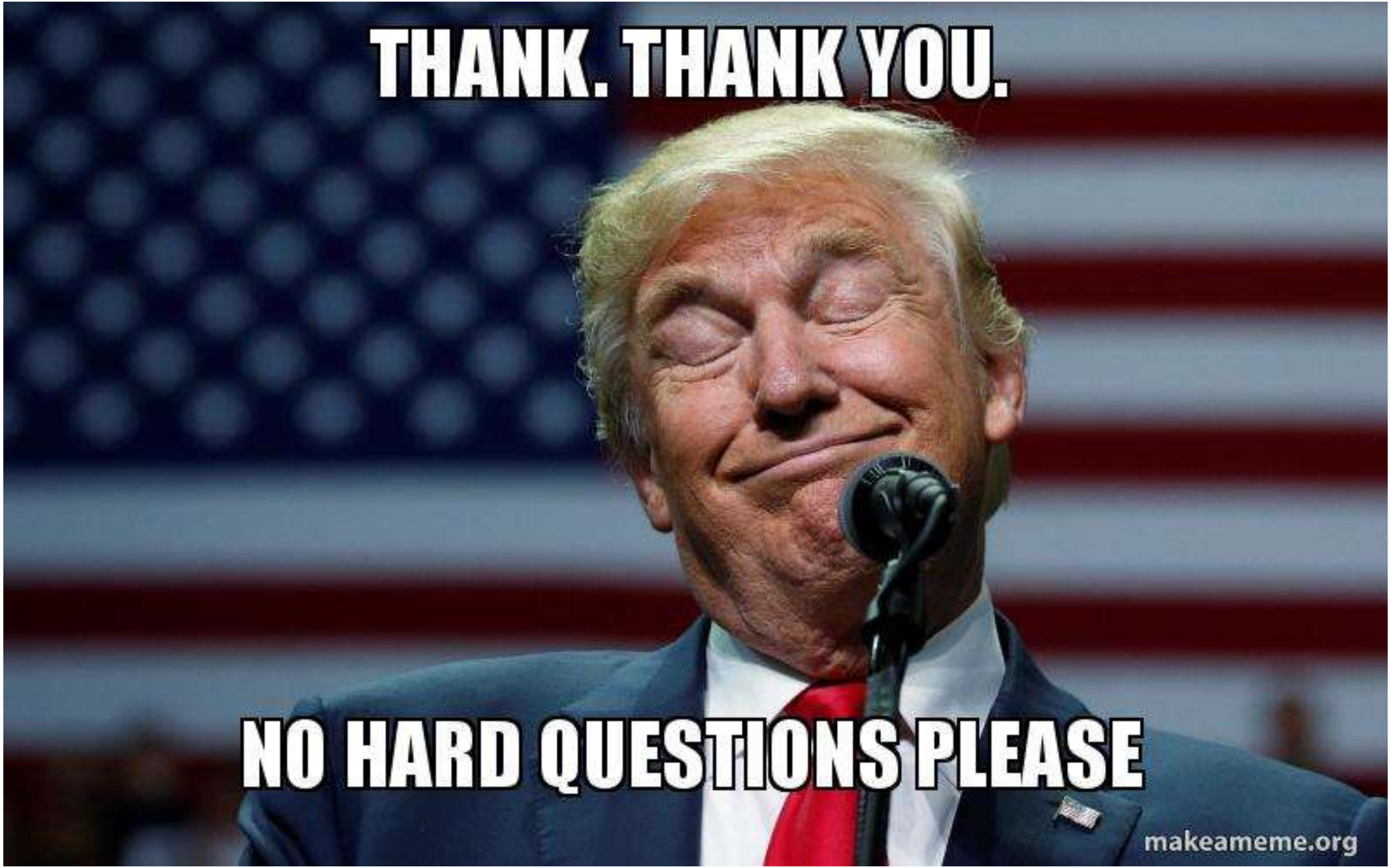
Table 7: CoNLL-2003 Named Entity Recognition results [1]

In Conclusion

- Bert major contribution was in adding more generalization to existing Transfer Learning methods by using bidirectional architecture.
- Bert model also added more contribution to the field. Fine-tuning Bert model will now tackle a lot of NLP tasks.

Papers that outperformed Bert

- Roberta, FAIR
- Xlnet, CMU + Google AI
- CTRL, Salesforce Research


THANK. THANK YOU.
NO HARD QUESTIONS PLEASE
makeameme.org

References

1. http://jalammar.github.io/illustrated-bert/
2. J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
3. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In NAACL.
4. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.
5. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.
6. Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. Computational linguistics, 18(4):467–479.
7. Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research, 6(Nov):1817–1853.

8. John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In Proceedings of the 2006 conference on empirical methods in natural language processing, pages 120–128. Association for Computational Linguistics.

9. 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.

10. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532– 1543.

11. Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL ’10, pages 384–394.

12. Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302.

13. Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In International Conference on Learning Representations.

14. Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In International Conference on Machine Learning, pages 1188–1196.

15. Yacine Jernite, Samuel R. Bowman, and David Sontag. 2017. Discourse-based objectives for fast unsupervised sentence representation learning. CoRR, abs/1705.00557.

16. http://jalammar.github.io/illustrated-bert/

17. Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In Advances in neural information processing systems, pages 3079–3087.

18. Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In ACL. Association for Computational Linguistics.

19. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP Workshop Black box NLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355.

20. Alexis Conneau, Douwe Kiela, Holger Schwenk, Lo¨ıc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

21. Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In NIPS.

22. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09.

23. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

24. Wilson L Taylor. 1953. Cloze procedure: A new tool for measuring readability. Journalism Bulletin, 30(4):415–433

25. 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.

26. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392.

27. Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In ACL.

28. Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP).

29. Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In CoNLL.

30. http://jalammar.github.io/illustrated-transformer/