Advanced Transformers:

Pre-trained Language Modeling

ELMo, BERT, GPT and beyond

Marta Ruiz Costa-jussà, José Adrián Rodríguez Fonollola and Noé Casas
Timeline

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Word2Vec</td>
<td>GloVe</td>
<td>FastText</td>
<td>Transformer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attention</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bahdanau et al.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELMO; BERT; GPT</td>
<td>XLM; BART</td>
<td>GPT-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peters; Devlin; Radford</td>
<td>Lample; Lewis</td>
<td>Brown et al.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Pre-trained Language Modeling
  - Motivation: contextual word embeddings
  - Background: LM and Transformers
  - ELMO, BERT, GPT
- Latest Model GPT-3
- Evaluation
Same word can have different meaning depending on the context. Example:

- *Please, type everything in lowercase.*
- *What type of flowers do you like most?*

Classic word embeddings offer the same vector representation regardless of the context.

Solution: create word representations that depend on the context.
Why Contextual Embeddings?

• Train model in one of multiple tasks that lead to word representations.

• Release pre-trained models.

• Use pre-trained models, options:
  
  A. Fine-tune model on final task.

  B. Directly encode token representations with model.
Explaining why graphically

Phase 1: semi-supervised training
- *LM task
- LM task head (projection + softmax)
- Language Modeling Architecture
- Contextual representations
- Monolingual corpus

Phase 2: downstream task fine-tuning
- Downstream task
- Downstream task head
- Language Modeling Architecture
- Transfer learning

*downstream* tasks is what the field calls those supervised-learning tasks that utilize a pre-trained model or component.

Explaining why graphically

Phase 1: semi-supervised training
- *LM task
- LM task head (projection + softmax)
- Language Modeling Architecture
- Contextual representations
- Monolingual corpus

Phase 2: downstream task fine-tuning
- Downstream task
- Downstream task head
- Language Modeling Architecture
- Transfer learning

*downstream* tasks is what the field calls those supervised-learning tasks that utilize a pre-trained model or component.
Background: Language Modeling

- Data: Monolingual Corpus
- Task: predict next token given previous tokens (causal):
  \[ P(T_i \mid T_1\ldots T_{i-1}) \]
- Usual models: LSTM, Transformer.
Non-recurrent attention-based language models

Convolution

Self-Attention

high attention

low attention

She is eating a green apple.
Transformer Language Model (Self-attention)

fig: Jordi armengol
“Masked language models” can be used for any NLP task as “contextual word embeddings”
<table>
<thead>
<tr>
<th>Model Alias</th>
<th>Org.</th>
<th>Article Reference</th>
</tr>
</thead>
</table>
| ELMo       | AllenNLP | *Deep contextualized word representations*  
Peters et al. |
| OpenAI GPT | OpenAI    | *Improving Language Understanding by Generative Pre-Training*  
Radford et al. |
| BERT       | Google   | *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*  
Devlin et al. |
## Differences

<table>
<thead>
<tr>
<th>Alias</th>
<th>Model</th>
<th>Token</th>
<th>Tasks</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMo</td>
<td>LSTM</td>
<td>word</td>
<td>Bidirectional LM</td>
<td>English</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>Transformer decoder</td>
<td>subword</td>
<td>Causal LM + Classification</td>
<td>English</td>
</tr>
<tr>
<td>BERT</td>
<td>Transformer encoder</td>
<td>subword</td>
<td>Masked LM + Next sentence prediction</td>
<td>Multilingual</td>
</tr>
</tbody>
</table>
**ELMO: contextual word representations**

- **Task:** bidirectional LM
- **Model:** 2-layer biLSTM
- **Tokens:** words
OpenAI GPT: Generative Pre-Training

- **Task:** causal LM
- **Model:** self-attention layers
- **Tokens:** subwords
- **Transformer Decoder**

Learning a generative language model using unlabeled data and then fine-tuning the model by providing examples of specific downstream tasks.
More on GPT

- Learning objectives and concepts:
  - unsupervised language modelling (pre-training)
    \[ L_1(T) = \sum_i \log P(t_i|t_{i-k}, \ldots, t_{i-1}; \theta) \] (i)
  - supervised fine-tuning and modified training objective
    \[ L_2(C) = \sum_{x,y} \log P(y|x_1, \ldots, x_n) \] (ii)
    \[ L_3(C) = L_2(C) + \lambda L_1(C) \] (iii)

- tasks like textual entailment, semantic similarity, question answering and commonsense reasoning,
**Tasks**: masked LM + next sentence prediction

**Model**: self-attention layers

**Tokens**: subwords

**Transformer Encoder**
Masked language modeling task

BERT is based on “gap-fill” exercises (Transformer)

Sherlock Holmes is probably the most famous detective in MASK. Of course, he wasn't a real person. His MASK is based on a real man whose career had a great influence on Arthur Conan Doyle, the author of the detective stories.

Mask out k% of the input words, and then predict the masked words
- They always use $k = 15\%$
- Too little masking: Too expensive to train
- Too much masking: Not enough context

https://arxiv.org/abs/1810.04805
Next sentence prediction task

- To learn relationships between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence.

<table>
<thead>
<tr>
<th>Sentence A</th>
<th>Sentence B</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>The man went to the store.</td>
<td>He bought a gallon of milk.</td>
<td>IsNextSentence</td>
</tr>
<tr>
<td>The man went to the store.</td>
<td>Penguins are flightless.</td>
<td>NotNextSentence</td>
</tr>
</tbody>
</table>
Contextual word embeddings. BERT

Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

https://arxiv.org/abs/1810.04805
Contextual word embeddings. BERT

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

https://arxiv.org/abs/1810.04805
Bert Zoo

Corpora/Data
- Youtube videos
- BooksCorpus
- Biomedical corpus
- Scientific publications
- SWAG, IMDB, Twitter
- Clinical notes/EHR
- English Wikipedia
- Monolingual Corpora (104 languages)
- Hierarchical diagnostic codes

Graph Neural Net

Pre-trained models
- SciBERT
- ERNIE (1)
- M-BERT
- ERNIE (2)
- G-BERT
- BERT
- ClinicalBERT
- TransBERT

Fine-tuned models
- PatentBERT
- DocBERT
- Video captioning (classification)
- Prediction tasks (e.g., Hospital readmission)
- Classification tasks (e.g., medication recommendation)
- Code Switching (e.g., English/Hindi mix sentences)
Summary: ELMO, GPT, BERT

<table>
<thead>
<tr>
<th>Alias</th>
<th>Model</th>
<th>Token</th>
<th>Tasks</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMo</td>
<td>LSTM</td>
<td>word</td>
<td>Bidirectional LM</td>
<td>English</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>Transformer</td>
<td>subword</td>
<td>Causal LM + Classification</td>
<td>English</td>
</tr>
<tr>
<td>BERT</td>
<td>Transformer</td>
<td>subword</td>
<td>Masked LM + Next sentence prediction</td>
<td>Multilingual</td>
</tr>
</tbody>
</table>
LATEST LANGUAGE MODELS
GPT-2 AND GPT-3
OpenAI GPT-2: language model and specific tasks

Just a really big Transformer LM
• Trained on 40GB of text
• Quite a bit of effort going into making sure the dataset is good quality
• Take webpages from reddit links with high karma

It does:

(1) Obviously, language modeling (but very well)!
(2) Zero-Shot Learning: no supervised training data!
• Ask LM to generate from a prompt
• Reading Comprehension: <context> <question> A:
• Summarization: <article> TL;DR:
• Translation:
  <English sentence1> = <French sentence1>
  <English sentence 2> = <French sentence 2>
  …..
  <Source sentence> =
• Question Answering: <question> A:
GPT-3: Language Models are Few-Shot Learners

- “For all tasks, GPT-3 is applied **without any gradient updates or fine-tuning**, with tasks and few-shot demonstrations specified purely via text interaction with the model.”

---

**Zero-shot learning:**

**Task description:** Convert English to French

**Prompt:**

- cheese =>

**One-shot learning:**

**Task description:** Convert English to French

**Example:**

- Sea-otter => loutre de mer
- Peppermint => menthe poivrée

**Prompt:**

- cheese =>

**Few-shot learning:**

**Task description:** Convert English to French

**Example:**

- Sea-otter => loutre de mer
- Peppermint => menthe poivrée

**Prompt:**

- cheese =>
GPT-3

- Parameters

![Bar chart comparing the number of parameters (B) for different models: BERT, RoBERTa, GPT-2, T5, Turing NLG, and GPT-3. GPT-3 has significantly more parameters compared to the other models.]
GPT-3 Applications

Chatbots

Human: Hey, how are you doing?
AI: I'm good! What would you like to chat about?
Human: Have you done any coding with Python?
AI: Yeah! I've done a lot of Python coding as it is my favorite language.
Human: Nice, it's my favorite too! What have you done with it?
AI: Oh, a lot. I have done everything from creating Python modules to creating micro-websites with it.
Human: What web framework do you use?
AI: I used to use Django, but it was too difficult for me. I have found Flask is a better option as it's lighter and simpler.
Human: Wow, I agree 100%. Have you used WebSocket at all?
AI: Yes, I actually have written a chatbot using it!
Human:

Email generation

Translation

**ENGLISH**

People who keep pet lizards terrify me.

**CHINESE (SIMPLIFIED)**

养宠物蜥蜴的人吓坏了我。

SQL-Prompt

```sql
SELECT
    avg(count)
FROM
    (SELECT
        user_id,
        count(*)
    FROM
        subscribers
    GROUP BY
        user_id)
GROUP BY
    user_id
```
EVALUATION
Evaluation

- **RACE**: Multiple choices on reading comprehension
- **SQuAD**: Extractive Question and Answering
- **CNN/DailyMail XSum**: Summarisation
- **GLUE**: Various tasks, including classification of sentence and sentence pair relationship, classification of text pair relationship, extractive question answering, and sentiment analysis
- **BLEU**: Machine translation
Example BERT for SQuAD (Stanford Question Answering Dataset)

Given a question and a paragraph from Wikipedia containing the answer, the task is to predict the answer text span in the paragraph. Example:

- **Input Question:**
  Where do water droplets collide with ice crystals to form precipitation?

- **Input Paragraph:**
  ... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. ...

- **Output Answer**
  within a cloud
Example BERT for SQuAD (Stanford Question Answering Dataset)

- Start with a pretrained BERT (‘gap-fill’ task) with BookCorpus and Wikipedia.
- Train BERT for SQuAD with an additional start vector $S$ and end vector $E$ using the SQuAD training data.

<table>
<thead>
<tr>
<th>System</th>
<th>Dev EM</th>
<th>Dev F1</th>
<th>Test EM</th>
<th>Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaderboard (Oct 8th, 2018)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>-</td>
<td>-</td>
<td>82.3</td>
<td>91.2</td>
</tr>
<tr>
<td>#1 Ensemble - nlnet</td>
<td>-</td>
<td>-</td>
<td>86.0</td>
<td>91.7</td>
</tr>
<tr>
<td>#2 Ensemble - QANet</td>
<td>-</td>
<td>-</td>
<td>84.5</td>
<td>90.5</td>
</tr>
<tr>
<td>#1 Single - nlnet</td>
<td>-</td>
<td>-</td>
<td>83.5</td>
<td>90.1</td>
</tr>
<tr>
<td>#2 Single - QANet</td>
<td>-</td>
<td>-</td>
<td>82.5</td>
<td>89.3</td>
</tr>
<tr>
<td>Published</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiDAF+ELMo (Single)</td>
<td>-</td>
<td>85.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R.M. Reader (Single)</td>
<td>78.9</td>
<td>86.3</td>
<td>79.5</td>
<td>86.6</td>
</tr>
<tr>
<td>R.M. Reader (Ensemble)</td>
<td>81.2</td>
<td>87.9</td>
<td>82.3</td>
<td>88.5</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT$_{BASE}$ (Single)</td>
<td>80.8</td>
<td>88.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT$_{LARGE}$ (Single)</td>
<td>84.1</td>
<td>90.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT$_{LARGE}$ (Ensemble)</td>
<td>85.8</td>
<td>91.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT$_{LARGE}$ (Sgl.+TriviaQA)</td>
<td>84.2</td>
<td>91.1</td>
<td>85.1</td>
<td>91.8</td>
</tr>
<tr>
<td>BERT$_{LARGE}$ (Ens.+TriviaQA)</td>
<td>86.2</td>
<td>92.2</td>
<td>87.4</td>
<td>93.2</td>
</tr>
</tbody>
</table>
Evaluation

GLUE
test set performance

# Parameters (M) in log scale
Evaluation

SQuAD 1.1
F1 score on dev set

# Parameters (M) in log scale

- ALBERT
- Electra-Large
- RoBERTa
- XLNet
- BART-Large
- T5-3B
- T5-Base
- Human
- MobileBERT
- Electra-small
- Bert-Large
- BERT-Base
- DistilBERT (dev set)
CLASSIFICATION

Encoder only
- BERT
- RoBERTa
- Reformer
- FlaubERT
- CamemBERT
- Electra*
- MobileBERT
- Longformer

Decoder only
- Transformer-XL
- XLNet
- GPT series
- DialoGPT

Encoder + Decoder
- Transformer
- XLM
- T5
- BART
- XLM-RoBERTa
- Pegasus
- mBART

## TRAINING TASKS

<table>
<thead>
<tr>
<th>Language task</th>
<th>Task specific head</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
<td></td>
</tr>
<tr>
<td>Sequence classification</td>
<td>Linear projection</td>
</tr>
<tr>
<td>Extractive question answering</td>
<td></td>
</tr>
<tr>
<td>Named entity recognition</td>
<td></td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td></td>
</tr>
<tr>
<td>Predict masked words</td>
<td>Masked language model</td>
</tr>
<tr>
<td>Predict next word</td>
<td>Causal language model</td>
</tr>
<tr>
<td><strong>Sentence</strong></td>
<td></td>
</tr>
<tr>
<td>Abstractive question answering</td>
<td>Causal language model or decoder</td>
</tr>
<tr>
<td>Summarisation</td>
<td></td>
</tr>
<tr>
<td>Machine translation</td>
<td></td>
</tr>
<tr>
<td>Text generation</td>
<td></td>
</tr>
</tbody>
</table>
Recycling is good for the world.

Recycling is NOT good for the world. It is bad for the environment, it is bad for our health, and it is bad for our economy. I'm not kidding. Recycling is not good for the environment. It is destructive to the earth and it is a major contributor to global warming. Recycling is not good for our health. It contributes to obesity and diseases like heart disease and cancer. Recycling is bad for our economy. It increases the cost of a product, and in turn, the price of everything that is made with that product. Recycling is not good for our nation. We pay a tremendous price for the privilege of having the world’s most advanced and efficient recycling system. Recycling is a huge, colossal waste of time, energy, money, and resources. And THAT is why we need to get back to basics and get back to basics in our recycling efforts. One of the best ways to start is to look at the process of creating a paper product. When you make a paper product, it is basically a long chain of materials. Everything from the raw materials (wood, cardboard, paper, etc.), to the reagents (dyes, solvents, etc.) to the printing equipment (chemicals, glue, paper, ink, etc.), to the packaging, to the packaging materials (mercury, chemicals, etc.) to the processing equipment (heating, cooling, etc.), to the packaging materials, to the packaging materials that are shipped overseas and to the packaging materials that are used in the United States. Each step along the way creates tons of waste that we constantly have to clean up. The process of making a paper product is a very wasteful one. But the end result is something that all of us need to consume. And if we want to keep the recycling process running efficiently, then we really need to think about each and every step that goes into making a paper product.
Question

- Dangerous language models?
Dangerous language models?

“GPT-3 has the potential to advance both the beneficial and harmful applications of language models.” — OpenAI Researchers

“Apart from consuming a massive amount of energy and impacting the environment, GPT-3 also comes with other challenges. With GPT-3 scraping down the whole internet archive to generate texts, it can heavily pose a threat to disinformation, where it can be used by bad actors to create an endless amount of fake news, spread misinformation amid COVID and carry out phishing scams. This could be easily attributed to the high-quality text generation capability that GPT-3 encompasses, making the texts convincingly human-like.” – Analytics Intia Mag
<table>
<thead>
<tr>
<th>model</th>
<th>title</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULMFiT</td>
<td>Universal Language Model Fine-tuning for Text Classification</td>
<td><a href="https://arxiv.org/abs/1801.06146">https://arxiv.org/abs/1801.06146</a></td>
</tr>
<tr>
<td>ELMo</td>
<td>Deep contextualized word representations</td>
<td><a href="https://arxiv.org/abs/1802.05365">https://arxiv.org/abs/1802.05365</a></td>
</tr>
<tr>
<td>model</td>
<td>title</td>
<td>link</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Bart</td>
<td>BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension</td>
<td>[<a href="https://arxiv.org/abs/1910.13461">https://arxiv.org/abs/1910.13461</a>]</td>
</tr>
<tr>
<td>DialogPT</td>
<td>DialogPT: Large-Scale Generative Pre-training for Conversational Response Generation</td>
<td>[<a href="https://arxiv.org/abs/1911.00536">https://arxiv.org/abs/1911.00536</a>]</td>
</tr>
<tr>
<td>XLM-RoBERTa</td>
<td>Unsupervised Cross-lingual Representation Learning at Scale</td>
<td>[<a href="https://arxiv.org/abs/1911.02116">https://arxiv.org/abs/1911.02116</a>]</td>
</tr>
<tr>
<td>Openai (GPT)</td>
<td>Improving Language Understanding by Generative Pre-Training</td>
<td>[<a href="https://openai.com/blog/language-unsupervised/">https://openai.com/blog/language-unsupervised/</a>]</td>
</tr>
<tr>
<td>GPT2</td>
<td>Language Models are Unsupervised Multitask Learners</td>
<td>[<a href="https://openai.com/blog/better-language-models/">https://openai.com/blog/better-language-models/</a>]</td>
</tr>
<tr>
<td>GPT3</td>
<td>Language Models are Few-Shot Learners</td>
<td>[<a href="https://arxiv.org/abs/2005.14165">https://arxiv.org/abs/2005.14165</a>]</td>
</tr>
</tbody>
</table>
Other Resources

- [https://nlp.stanford.edu/seminar/details/jdevlin.pdf](https://nlp.stanford.edu/seminar/details/jdevlin.pdf)
- [https://medium.com/dissecting-bert/dissecting-bert-part2-335ff2ed9c73](https://medium.com/dissecting-bert/dissecting-bert-part2-335ff2ed9c73)
- [https://github.com/huggingface/pytorch-pretrained-BERT](https://github.com/huggingface/pytorch-pretrained-BERT)
- [https://www.geeksforgeeks.org/open-ai-gpt-3/](https://www.geeksforgeeks.org/open-ai-gpt-3/)