Which *BERT?
A Survey Organizing Contextualized Encoders

Patrick Xia    Shijie Wu    Benjamin Van Durme
BERT is a 12 (or 24) layer Transformer language model trained on two pretraining tasks, masked language modeling (fill-in-the-blank) and next sentence prediction (binary classification), and on English Wikipedia and BookCorpus.

About BERT and friends

Why this size and architecture? -base, -large, -small, -xl, etc.

Why these tasks?XLNet, ELECTRA, SpanBERT, LXMERT, etc.

Why these tasks?masked language modeling (fill-in-the-blank)

How much “language”?linguistic probing tasks, attention, few-shot evaluation

What’s special about this data?BioBERT, Covid-Twitter-BERT, etc

Other languages?mBERT, XLM, XLM-R, mT5, RuBERT, etc
Using *BERTs

• Pretrain → finetune
  • *Pretrain* encoders on pretraining tasks (high-resource/data, possibly unsupervised)
  • *Finetune* encoders on target task (low-resource, expensive annotation)

• Primary method of evaluation: Natural Language “Understanding” (NLU)
  • Question Answering and Reading Comprehension
  • Commonsense
  • Textual Entailments
The story so far...

Pretraining | Efficiency | Data | Interpretability | Multilinguality
Pretraining

- Quantitative improvements in downstream tasks are made through pretraining methods

**Predict tokens in text**
- Masked language modeling
  - Masked token/word/span prediction
  - Replaced word prediction

**Predict other signals**
- Next sentence/segment prediction
- Discourse relations
- Grounding
  - To KB
  - Visual/multimodal
Efficiency

- **Training:**
  - Faster convergence with improved optimizers, hardware

- **Inference size/time:**
  - Large $\rightarrow$ small: knowledge distillation, pruning
  - Start small: parameter sharing/factorization, quantization

- **Are these techniques compared equally?**
  - Do we care about %parameter reduction? Memory? Inference time?
    These don’t necessarily correlate
  - Do we care about all tasks or just downstream one(s)?
Data

• Quantity: more data is better
  • Are comparisons across encoders fair?

• Quality: clean, *in-domain*, data is better
  • What are our test sets?

• Where is our data coming from??
  • Do we know what biases the contextualized encoders learn?
  • Should we use biased model in real systems?
Interpretability

• Task probing
  → Finetune pretrained models to test specific linguistic phenomenon

• Model weight inspection
  → Visualize weights for important words in input or layers in model

• Input Prompting
  → Force language models to fill in or complete text

• None of these methods are perfect
  • Task probing: more finetuning
  • Weight inspection: not reliable
  • Prompting: picking the prompt is critical
Multilinguiality

• A single encoder on multilingual text with shared input vocabulary

• These models do well! Why?
  • Shared vocabulary
  • Shared (upper) layers
  • Deep networks
  • Embeddings across languages can be aligned

• When are multilingual models good?
Shortcomings

Leaderboards | Overfitting our understanding | Expensive evaluations
Shortcomings

• Leaderboards without a leader
  • Publish & publicize negative submitted results
  • Leaderboard owners can periodically survey submissions?

• Overfitting our understanding
  • Interpretability/probing studies look at default pretrained models
  • Draw more conclusions across models in addition to across tasks

• Expensive evaluations
  • How can we make evaluation easier?
  • Unit testing?
So, which *BERT?

What is your ... task | data | language | goal?
What is your task?

• Not all tasks benefit from the shiniest encoder!
  • Some pretrained systems work well with just BERT
  • Encodings are just inputs to complex systems that are further tuned
• Finetuning and retraining entire models may not be feasible or even justified for your task
What is your data?

• Does the domain of your data overlap with that of the encoder?
• Is there are specialized pretrained encoder for your domain or data?
• Do you have enough data to train your own?
• Do you even need contextualized encoders?
What is your language?

- Is your language low-resource?
  - Use the best general-purpose model
  - Again, depends on your task and data

- Is there a competitive monolingual contextualized encoder?
  - Chinese, French, etc
  - Monolingual data curation may be better
  - Language-specific model hyperparameters can be adjusted (e.g. vocabulary)
What is your goal?

• Encoder research?
  • Build off great recent ideas
  • Incorporate “beta” and “nightly” ideas!

• Product development, fast deployment, something that works?
  • Pick well-documented models
  • HuggingFace Transformers uses a single interface; models can be easily upgraded later
Summary

• Contextualized encoders have transformed research and thinking in NLP in just a couple years
• Areas we are focusing on:
  • Pretraining, efficiency, data, interpretability, and multilinguality
• Are we making progress?
• Which model should you use?
  • Depends on task, data, language, and objective