Which *BERT?
A Survey Organizing Contextualized Encoders

Patrick Xia  Shijie Wu  Benjamin Van Durme
Background

History of Text Representations | Who is BERT? | About BERT and friends
A History of Text Representations

• Co-occurrence statistics
  • Brown Clusters
  • Count vectors, TF-IDF vectors, co-occurrence matrix decomposition

• Predictive
  • word2vec, GloVe, CBOW, Skip-Gram, etc

• Contextualized language models
  • Representation of word *changes* based on context
  • CoVE, ELMo, GPT, BERT, etc
A History of Text Representations

• Co-occurrence statistics
  • Brown Clusters
  • Count vectors, TF-IDF vectors, co-occurrence matrix decomposition

• Predictive
  • word2vec, GloVe, CBOW, Skip-Gram, etc

• Contextualized language models
  • Representation of word *changes* based on context
  • CoVE, ELMo, GPT, BERT, etc
A History of Text Representations

• Co-occurrence statistics
  • Brown Clusters
  • Count vectors, TF-IDF vectors, co-occurrence matrix decomposition

• Predictive
  • word2vec, GloVe, CBOW, Skip-Gram, etc

• Contextualized language models
  • Representation of word changes based on context
  • CoVE, ELMo, GPT, BERT, etc
Who is BERT? 🐦

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 BooksCorpus.
About BERT and friends

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

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

Why these tasks?
- masked language modeling (fill-in-the-blank)
- next sentence prediction (binary classification)

XLNet, ELECTRA, SpanBERT, LXMERT, etc.

Other languages?
mBERT, XLM, XLM-R, mT5, RuBERT, etc

BERT is a 12 (or 24) layer Transformer language model trained on

Wikipedia and BooksCorpus

What's special about this data?
- BioBERT, Covid-Twitter-BERT, etc
Using these *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
Using these *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
Using these *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
Using these *BERTs

• Pretrain $\rightarrow$ 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
Using these *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
Using these *BERTs

• Pretrain $\rightarrow$ 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
Using these *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
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
Pretraining

- Masked language modeling

What is natural language processing?

What is natural language [mask]?

What is [mask] processing?

Who is natural language processing?

Masked token prediction

Masked span prediction

Replaced word prediction
Pretraining

• Predict other signals
  • Next “sentence” prediction
  • Discourse markers/relations
  • Real-world knowledge: knowledge base/IR scores
  • Visual and multimodal grounding
Efficiency

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

• **Inference size/time:**
  • Large $\to$ 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)?
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)?
Efficiency

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

• **Inference size/time:**
  - Large $\to$ 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)?
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)?
Efficiency

• Training:
  • Faster convergence with improved optimizers, hardware

• Inference size/time:
  • Large → 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?
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?
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?
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?
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?
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?
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
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
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
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
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
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
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
Multilinguality

- 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?
Multilinguality

• 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?
Multilinguality

- 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?
Multilinguality

• 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?
Multilinguality

• 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?
Multilinguality

• 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?
Multilinguality

• 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
Leaderboards without a leader

• There’s good science in 2nd place: who is responsible for publishing when reviewers demand 1st?
  • Publicize and publish negative results

• Leaderboard owners should be responsible for frequently surveying submissions

• A leaderboard is not just a dataset: it’s an unending shared task
  • Shared tasks need summaries discussing all the methods
Leaderboards without a leader

- There’s good science in 2nd place: who is responsible for publishing when reviewers demand 1st?
  - Publicize and publish negative results
- Leaderboard *owners* should be responsible for frequently surveying submissions
- A leaderboard is not just a dataset: it’s an unending shared task
  - Shared tasks need summaries discussing all the methods
Leaderboards without a leader

• There’s good science in 2\textsuperscript{nd} place: who is responsible for publishing when reviewers demand 1\textsuperscript{st}?
  • Publicize and publish negative results

• Leaderboard *owners* should be responsible for frequently surveying submissions

• A leaderboard is not just a dataset: it’s an unending shared task
  • Shared tasks need summaries discussing all the methods
Leaderboards without a leader

• There’s good science in 2\textsuperscript{nd} place: who is responsible for publishing when reviewers demand 1\textsuperscript{st}?
  • Publicize and publish negative results

• Leaderboard \textit{owners} should be responsible for frequently surveying submissions

• A leaderboard is not just a dataset: it’s an unending shared task
  • Shared tasks need summaries discussing all the methods
Leaderboards without a leader

• There’s good science in 2nd place: who is responsible for publishing when reviewers demand 1st?
  • Publicize and publish negative results

• Leaderboard *owners* should be responsible for frequently surveying submissions

• A leaderboard is not just a dataset: it’s an unending shared task
  • Shared tasks need summaries discussing all the methods
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across *models* in addition to across *tasks*
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across models in addition to across tasks
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across models in addition to across tasks
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across *models* in addition to across *tasks*
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across *models* in addition to across *tasks*
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across *models* in addition to across *tasks*
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Overfitting our understanding

• We know so much about English Wikipedia + BooksCorpus, 12-layer and 24-layer BERT.
  • What about 8-layer BERT? Or distilled BERT?
  • Simple English Wikipedia + RoBERTa?
  • BooksCorpus (subgenres) + XLNet?

• Draw more conclusions across *models* in addition to across *tasks*
  • At what point (#params) does a model outperform humans on X?
  • How much of Wikipedia does a model need to outperform on Y?
Expensive evaluations

• GLUE: 9 tasks, SuperGLUE: 10 tasks, SQuAD: 150K QA pairs
  • Finetuning cost for every task is high
• If a researcher is focused on distilling encoders with several novel methods:
  • Finetune all tasks $\Rightarrow$ unrelated effort and time running “evaluation”
  • Stick with a few tasks $\Rightarrow$ unfair comparisons, angry reviewers
• How can we make evaluation easier?
• Unit testing models for practical applications?
Expensive evaluations

• GLUE: 9 tasks, SuperGLUE: 10 tasks, SQuAD: 150K QA pairs
  • Finetuning cost for every task is high
• If a researcher is focused on distilling encoders with several novel methods:
  • Finetune all tasks → unrelated effort and time running “evaluation”
  • Stick with a few tasks → unfair comparisons, angry reviewers
• How can we make evaluation easier?
• Unit testing models for practical applications?
Expensive evaluations

• GLUE: 9 tasks, SuperGLUE: 10 tasks, SQuAD: 150K QA pairs
  • Finetuning cost for every task is high
• If a researcher is focused on distilling encoders with several novel methods:
  • Finetune all tasks → unrelated effort and time running “evaluation”
  • Stick with a few tasks → unfair comparisons, angry reviewers
• How can we make evaluation easier?
• Unit testing models for practical applications?
Expensive evaluations

- GLUE: 9 tasks, SuperGLUE: 10 tasks, SQuAD: 150K QA pairs
  - Finetuning cost for every task is high
- If a researcher is focused on distilling encoders with several novel methods:
  - Finetune all tasks → unrelated effort and time running “evaluation”
  - Stick with a few tasks → unfair comparisons, angry reviewers
- How can we make evaluation easier?
- Unit testing models for practical applications?
Expensive evaluations

- GLUE: 9 tasks, SuperGLUE: 10 tasks, SQuAD: 150K QA pairs
  - Finetuning cost for every task is high
- If a researcher is focused on distilling encoders with several novel methods:
  - Finetune all tasks \(\rightarrow\) unrelated effort and time running “evaluation”
  - Stick with a few tasks \(\rightarrow\) unfair comparisons, angry reviewers

- How can we make evaluation easier?
- Unit testing models for practical applications?
Expensive evaluations

- GLUE: 9 tasks, SuperGLUE: 10 tasks, SQuAD: 150K QA pairs
  - Finetuning cost for every task is high
- If a researcher is focused on distilling encoders with several novel methods:
  - Finetune all tasks → unrelated effort and time running “evaluation”
  - Stick with a few tasks → unfair comparisons, angry reviewers
- How can we make evaluation easier?
- Unit testing models for practical applications?
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 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 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 a 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 data?

- Does the domain of your data overlap with that of the encoder?
- Is there a 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 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 data?

• Does the domain of your data overlap with that of the encoder?
• Is there a 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 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 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 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 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 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
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
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
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
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
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
Thank you

Please join the Q&A for discussion

See paper for more details and references