Deep Contextual Neural Word Representations: Linguistic Structure Discovery and Efficient Discriminative Training

Christopher Manning
Stanford University and CIFAR Fellow
@chrmanning 🌸 @stanfordnlp
ElementAI/MILA, December 2019 (last talk of 2019!)
Plan

1. From recurrent sequence models to BERT transformers
2. BERT as a linguistic structure discovery machine
3. More efficient Discriminative Pre-training of Text Encoders
1. Language Modeling

A Language Model (LM) predicts a word in a context:

\[
\text{the students opened their ______} \\
\text{books} \quad \text{laptops} \quad \text{exams} \quad \text{minds}
\]

An LM is a key part of decoding tasks like speech recognition, spelling correction, and any NL generation task, including machine translation, summarization, and story generation.
LMs in The Dark Ages: \( n \)-gram models

Count how often words follow word sequences; divide to get cond. prob.

Classic curse of dimensionality scenario: zillions of params

Markov assumption:
\[
P(x^{(t+1)}|\text{President Trump denied the}) \approx P(x^{(t+1)}|\text{denied the})
\]

Discounting/Smoothing

\[
P_{bo}(x^{(3)}|x^{(2)}, x^{(1)}) \approx \lambda P(x^{(3)}|x^{(2)}, x^{(1)}) + (1 - \lambda) P(x^{(3)}|x^{(2)})
\]

Mixture/Backoff
How much of the intricate structure of human languages do these language models know?

- (Passionately argued!) answer of linguists: almost none
  - Though they know quite a bit of simple world knowledge
    - The ship {sailed, sank, anchored, ...}
  - And, in an unaggregated way, they know some low-level syntax
    - They know you tend to get sequences like:
      - preposition – article – noun
      - article – adjective – noun
  - But they don’t know the concept “noun” or sentence structure rules
    - As an abstracted grammar
Capturing conventional linguistics in NLP

Part-of-Speech:

Basic Dependencies:

Coreference:
The Faculty of Language: What Is It, Who Has It, and How Did It Evolve?

Marc D. Hauser,¹* Noam Chomsky,² W. Tecumseh Fitch³

We argue that an understanding of the faculty of language requires substantial interdisciplinary cooperation. We suggest how current developments in linguistics can be profitably wedded to work in evolutionary biology, anthropology, psychology, and neuroscience. We submit that a distinction should be made between the faculty of language in the broad sense (FLB) and in the narrow sense (FLN). FLB includes a sensory-motor system, a conceptual-intentional system, and the computational mechanisms for recursion, providing the capacity to generate an infinite range of expressions from a finite set of elements. We hypothesize that FLN only includes recursion and is the only uniquely human component of the faculty of language. We further argue that FLN may have evolved for reasons other than language, hence comparative studies might look for evidence of such computations outside of the domain of communication (for example, number, navigation, and social relations).

If a martian graced our planet, it would be struck by one remarkable similarity among Earth’s living creatures and a key difference. Concerning similarity, it would note that all
Enlightenment era neural language models (NLMs)

1. **Solve curse of dimensionality** by sharing of statistical strength via dense, low-dimensionality word vectors $v_1, v_2, \ldots, v_K$ [Bengio, Ducharme, Vincent & Jauvin JMLR 2003], etc.:

$$ P(x^{(t+1)}|x^{(t)}, x^{(t-1)}) = \text{softmax}(\text{FFNN}(v^{(t)}, v^{(t-1)})) $$

2. **Solve failure to exploit long contexts** via recurrent NNs

First, simple RNNs, soon usually LSTMs [Zaremba et al. 2014]

*the same stump which had impaled the car of many a guest in the past thirty years and which he refused to have removed*

$$ P(x^{(t+1)}|x^{\leq t}) = \text{LSTM}(h^{(t)}, x^{(t)}) $$
Flashback to 2017

The BiLSTM Hegemony

To a first approximation, the de facto consensus in NLP in 2017 is that no matter what the task, you throw a BiLSTM at it, with attention if you need information flow.
An LSTM encoder-decoder network

[Sutskever et al. 2014]
A BiLSTM encoder and LSTM-with-attention decoder
Progress in Machine Translation

[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]

2018 NLP breakthrough with big language models

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMo</td>
<td>Jan 2018</td>
<td>103M words</td>
</tr>
<tr>
<td>ULMfit</td>
<td>Jan 2018</td>
<td>103M words</td>
</tr>
<tr>
<td>GPT</td>
<td>June 2018</td>
<td>800M words</td>
</tr>
<tr>
<td>BERT</td>
<td>Oct 2018</td>
<td>3.3B words</td>
</tr>
<tr>
<td>GPT-2</td>
<td>Feb 2019</td>
<td>40B words</td>
</tr>
<tr>
<td>XL-Net, ERNIE</td>
<td>July 2019–</td>
<td></td>
</tr>
</tbody>
</table>

All these models are Transformer models.
Transformer (Vaswani et al. 2017)  
BERT (Devlin et al. 2018)
Transformer (Vaswani et al. 2017)
BERT (Devlin et al. 2018)

BERT (Bidirectional Encoder Representations from Transformers): Pre-training of Deep Bidirectional Transformers for Language Understanding, which is then fine-tuned for a particular task

Pre-training uses a cloze task formulation where 15% of words are masked out and predicted:

```
store                    gallon
↑                         ↑
the man went to the [MASK] to buy a [MASK] of milk
```

16
BERT model

Pre-train contextual word vectors in a LM-like way with transformers.

Learn a classifier built on the top layer for each task that you fine tune for.
Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi’s Stadium in the San Francisco Bay Area at Santa Clara, California.

**Question:** Which team won Super Bowl 50?
Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Question: Which team won Super Bowl 50?
Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

**Question:** Which team won Super Bowl 50?
My talk at the Automated Knowledge Base Construction (AKBC) workshop 2013

Texts are Knowledge
AllenAI ARISTO: Answering Science Exam Questions

Which equipment will best separate a mixture of iron filings and black pepper? (1) magnet (2) filter paper (3) triplebeam balance (4) voltmeter

Which process in an apple tree primarily results from cell division? (1) growth (2) photosynthesis (3) gas exchange (4) waste removal

<table>
<thead>
<tr>
<th>Test Set</th>
<th>IR</th>
<th>TupInf</th>
<th>Multee</th>
<th>AristoBERT</th>
<th>AristoRoBERTa</th>
<th>ARISTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regents 4th</td>
<td>64.5</td>
<td>63.5</td>
<td>69.7</td>
<td>86.2</td>
<td>88.1</td>
<td>89.9</td>
</tr>
<tr>
<td>Regents 8th</td>
<td>66.6</td>
<td>61.4</td>
<td>68.9</td>
<td>86.6</td>
<td>88.2</td>
<td>91.6</td>
</tr>
<tr>
<td>Regents 12th</td>
<td>41.2</td>
<td>35.4</td>
<td>56.0</td>
<td>75.5</td>
<td>82.3</td>
<td>83.5</td>
</tr>
<tr>
<td>ARC-Challenge</td>
<td>0.0</td>
<td>23.7</td>
<td>37.4</td>
<td>57.6</td>
<td>64.6</td>
<td>64.3</td>
</tr>
</tbody>
</table>
Google web search

BERT brings big gains to web search

Google is improving 10 percent of searches by understanding language context

Say hello to BERT

By Dieter Bohn | @backlon | Oct 25, 2019, 3:01am EDT

Google is currently rolling out a change to its core search algorithm that says could change the rankings of results for as many as one in ten queries. It’s based on cutting-edge natural language processing (NLP) techniques developed by Google researchers and
2. What does BERT know? Observational evidence

Kevin Clark, Urvashi Khandelwal, Omer Levy, & Christopher Manning (BlackBoxNLP 2019 workshop at ACL 2019 best paper)

• BERT works really well and calculates clearly useful context-dependent word representations

• Directly observe what BERT is looking at

• We find that BERT induces a lot of structure similar to conventional linguistic structure … because it helps predict
BERT Attention Heads

- For each of many attention heads, for each word position, see where BERT pays attention

- Look at the most-attended-to word for each head

- How does what BERT attends to correspond to linguistics?
What do BERT attention heads do?

1-1: Attend broadly (“BoW head”)

3-1: Attend to next (or prev) word
First layer heads mainly average

![Graph showing entropy vs. layer, with BERT heads and uniform attention highlighted.](image)
A sentence’s meaning is composed via its syntax tree

The chef that ran to the store was out of food

"the store was out of food" would be a valid sentence by itself
Does some of BERT attention resemble dependency syntax?

I went to the store

Take the most-attended-to words

Compare with dependency tree
A bunch of heads specialize on a syntactic relation (!)

Head 8-10
Direct objects attend to verbs
86.8% on dobj relation

Head 8-11
Noun modifiers (det, adj) attend to head noun. 94.3% on det relation

Overall, a combination of these heads can give an okay dependency parser: 77 UAS (Cf. 26 from right branching, 58 from GloVe word vecs + distance.)
BERT attention heads capture many dependency relations remarkably well

<table>
<thead>
<tr>
<th>Relation</th>
<th>Best head’s accuracy</th>
<th>Best baseline’s accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>35</td>
<td>26</td>
</tr>
<tr>
<td>pobj</td>
<td>76</td>
<td>35</td>
</tr>
<tr>
<td>det</td>
<td>94</td>
<td>52</td>
</tr>
<tr>
<td>dobj</td>
<td>87</td>
<td>40</td>
</tr>
<tr>
<td>poss</td>
<td>81</td>
<td>48</td>
</tr>
<tr>
<td>auxpass</td>
<td>83</td>
<td>41</td>
</tr>
</tbody>
</table>
Coreferent mentions attend to their antecedent; for not a mention words: no-op attention 85% on [SEP]. Head 5-4: **65.1%** accuracy at linking to head of antecedent
Cf. vs. 69% for a 4-sieve, rule-based system (cf. Lee et al. 2011)
choosing nearest {full string, headword, PNG match; any NP}
Experimental evidence

Hewitt and Manning (NAACL 2019)

tl;dr

Does BERT encode syntax (dependency trees) in its contextual representations?

Yes, approximately

How can we tell whether its vector representations encode trees?

Using a **structural probe** to look at the geometry
Are vector spaces and trees reconcilable?

- Are the vector space representations in NLP reconcilable with the discrete syntactic tree structures hypothesized for language?
Distance metrics unify trees and vectors

An undirected tree defines a distance metric on pairs of words, the path metric: the number of edges in the path between the words.

The edges of the tree can be recovered by looking at all distance=1 pairs.
Finding trees in vector spaces

We can look for trees in the vector space by looking for their **distances** and **norms** in the space.

Here's a sentence embedded by a NN!

\[ h_i, h_j : \text{vector representation of words } i \text{ and } j. \]
Finding trees in vector spaces

We don't expect all dimensions of the vector space to encode syntax -- NNs have a lot to encode!

We find the linear transformation that encodes syntax best.

B : The syntax transformation matrix
B_{h_i} : Syntax-transformed vector word representation
Finding trees in vector spaces

In the transformed space, (squared) L2 distance approximates tree distance.

\[ d_{\text{path}}(i,j) : \text{Tree path distance} \]
\[ \|B(h_i - h_j)\|_2^2 : \text{Squared Vector space distance (}\|h_i - h_j\|_B^2) \]

The
who
was
store
to
chef
food
out
ran
of
Finding trees in vector spaces

With this property, a minimum spanning tree in the vector space distance recovers the tree.
Does BERT encode undirected parse trees -> does there exist a distance transformation?

\[
\text{arg min}_B \sum_{\ell \in \text{PTB}} \frac{1}{|s^{\ell}|^2} \sum_{i,j} |d_{\text{path}}^{\ell}(i,j) - \|B(h_i^{\ell} - h_j^{\ell})\|^2_2|
\]

- Find a single transformation \(B\)
- over all word pairs in each sentence
- The difference between tree distance and squared vector distance is minimized
- such that over all sentences in PTB training
Trees are encoded well in these representations.
Legend:  
- far
- close

BERT structural probe

Gold parse tree

words
Trees from structural probe parse distances approximate parse trees pretty well!

Black (above sentence): Human-annotated parse tree
Teal (below sentence): Minimum spanning tree, structural probe on BERT

The complex financing plan in the S+L bailout law includes raising $30 billion from debt issued by the newly created RTC
Syntax geometry is quite low rank
Visualizing and Measuring the Geometry of BERT

[Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg, NeurIPS 2019]

https://pair-code.github.io/interpretability/bert-tree/

- What does syntax geometry look like?
- Why are trees encoded in \textit{squared} vector distance?
- Geometry + structural probes for understanding BERT syntax
- Representation of word senses in BERT
Visualizing and Measuring the Geometry of BERT

“Factories booked $236.74 billion in orders in September, nearly the same as the $236.79 billion in August, the Commerce Department said.”
Why are trees encoded in *squared* vector distance?

Nodes in trees have a natural vector embedding.

1. Assign edges orthogonal unit embeddings.

[Coenen et al., 2019]; https://pair-code.github.io/interpretability/bert-tree/
Why are trees encoded in *squared* vector distance? Nodes in trees have a natural vector embedding.

1. Assign edges orthogonal unit embeddings.
2. Assign each edge a direction (say, root→leaf)
3. Assign each node sum of embeddings of edges pointing “towards” it

\[
f(t_i) = e_1 + e_3 = (1, 0, 1, 0, 0, 0)
\]

[Coenen et al., 2019]; https://pair-code.github.io/interpretability/bert-tree/
Why are trees encoded in squared vector distance?  

Squared L2 distance preserves tree distances

\[ f(t_3) = \]
\[ e_1 + e_3 = \\
(1, 0, 1, 0, 0, 0) \]

\[ f(t_5) = \]
\[ e_1 + e_4 + e_6 = \\
(1, 0, 0, 1, 0, 1) \]

\[ f(t_5) - f(t_6) = \]
\[ e_3 - e_4 - e_6 = \\
(0, 0, 1, -1, 0, -1) \]
\[ \| f(t_5) - f(t_6) \|^2 = 3 \]

[Coenen et al., 2019]; https://pair-code.github.io/interpretability/bert-tree/
Why are trees encoded in squared vector distance?

You can’t isometrically embed tree distance in Euclidean space.

You can encode it in a “Pythagorean embedding”

\( f : M \rightarrow \mathbb{R}^n \) is a Pythagorean embedding if for all \( x, y \in M \),
\[ d(x, y) = \|f(x) - f(y)\|^2 \]
3. Electra: Efficient Discriminative Pre-training of Text Encoders

- Kevin Clark and Christopher Manning
Rapid Progress from Pre-Training (GLUE benchmark)

Over 3x reduction in error in 2 years, “superhuman” performance
But let’s change the x-axis to compute ...

BERT-Large uses 60x more compute than ELMo
But let’s change the x-axis to compute ...
More compute, more better?

ALBERT uses 10x more compute than RoBERTa
Language Model Pretraining

- ULMFit, ELMo, GPT, 

The artist sold the painting.
Masked Language Model Pretraining

- BERT, XLNet, RoBERTa, ...
Masked Language Model Pretraining

- Bidirectional gives better performance
Masked Language Model Pretraining

- Bidirectional gives better performance
- But less efficient because only learn from 15% of tokens per example
- Our method: best of both worlds
New Pre-Training Task: Replaced Token Detection

- Instead of [MASK], replace tokens with plausible alternatives

the artist sold the painting
New Pre-Training Task: Replaced Token Detection

• Instead of [MASK], replace tokens with plausible alternatives

the painter
the artist
sold
the car
painting
New Pre-Training Task: Replaced Token Detection

[Diagram showing connections between words: the, painter, sold, the, car]
New Pre-Training Task: Replaced Token Detection

![Diagram showing original and replaced tokens in a sentence: the painter sold the car.](image)
ELECTRA: Efficiently Learning an Encoder to Classify Token Replacements Accurately

Bidirectional model but learn from all tokens

Clark, Luong, Le, and Manning (2020)
Generating Replacements

Plausible alternatives come from small masked language model (the “generator”) trained jointly with ELECTRA

```
the → [MASK]  → sample  → the  → original
artist → artist  → artist  → original
sold → sold  → sold  → original
the → the  → the  → original
painting → [MASK]  → car  → replaced
```

MLM Loss

Binary classification loss
Results: Glue Score vs Compute

- ELMo
- GPT
- BERT-Base
- BERT-Large
- XLNet
- RoBERTa
- EL-Small
- EL-Base
- EL-Large
- EL-Large (100k steps)
- GloVe

Pre-Train FLOPs
### GLUE Results: ELECTRA-Small and smaller and smaller

<table>
<thead>
<tr>
<th>Model</th>
<th>Train/Infer Speedup over BERT-Base</th>
<th>GLUE Score</th>
<th>Train time / hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELMo</td>
<td>19x / 1.2x</td>
<td>71.2</td>
<td>14d on 3 1080s</td>
</tr>
<tr>
<td>ELECTRA 6.25%</td>
<td>722x / 8x</td>
<td>74.1</td>
<td>6h on 1 V100</td>
</tr>
<tr>
<td>BERT-Small (ours)</td>
<td>45x / 8x</td>
<td>75.1</td>
<td>4d on 1 V100</td>
</tr>
<tr>
<td>ELECTRA 25%</td>
<td>181x / 8x</td>
<td>77.7</td>
<td>1d on 1 V100</td>
</tr>
<tr>
<td>DistilBERT</td>
<td>- / 2x</td>
<td>77.8</td>
<td></td>
</tr>
<tr>
<td>GPT</td>
<td>1.6x / 1x</td>
<td>78.8</td>
<td></td>
</tr>
<tr>
<td>ELECTRA-Small</td>
<td>45x / 8x</td>
<td>79.0</td>
<td>4d on 1 V100</td>
</tr>
<tr>
<td>BERT-Base</td>
<td>1x / 1x</td>
<td>82.2</td>
<td>4d on 16 TPUv3s</td>
</tr>
</tbody>
</table>
SQuAD 2.0 dev Results: ELECTRA-Large

- BERT-Large architecture, trained on XLNet data

<table>
<thead>
<tr>
<th>Model</th>
<th>Train FLOPs</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>0.3x</td>
<td>81.8</td>
</tr>
<tr>
<td>XLNet</td>
<td>1.3x</td>
<td>88.8</td>
</tr>
<tr>
<td>RoBERTa (100k steps)</td>
<td>0.9x</td>
<td>87.7</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>4.5x</td>
<td>89.4</td>
</tr>
<tr>
<td>BERT-large (ours)</td>
<td>1x</td>
<td>87.5</td>
</tr>
<tr>
<td>ELECTRA</td>
<td>1x</td>
<td>89.6</td>
</tr>
</tbody>
</table>
### Efficiency Ablations: All-Tokens MLM

<table>
<thead>
<tr>
<th>Model</th>
<th>GLUE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>82.2</td>
</tr>
<tr>
<td>Replace MLM</td>
<td>82.4</td>
</tr>
<tr>
<td>ELECTRA 15%</td>
<td>82.4</td>
</tr>
<tr>
<td>All-Tokens MLM</td>
<td>84.3</td>
</tr>
<tr>
<td>ELECTRA</td>
<td>85.0</td>
</tr>
</tbody>
</table>

Clark, Luong, Le, and Manning (2020)
Electra

- Recent pre-training methods let models benefit from unprecedented compute scale
  - But our environment/energy use doesn’t benefit!
  - It is important to be sensitive to compute when reporting results

- Replaced token detection is a more effective pre-training task then masked language modeling
  - Can provide good results on a single GPU in hours/days
  - At larger scale, trains over 4x faster
Final thoughts

- Self-supervised (or “unsupervised”) learning is very successful for doing natural language understanding tasks
  - More successful than multi-task learning (if only because of data supply)
- However, one key limitation has been the size/cost of models
- Was annotating lots of linguistic data all a mistake?
  - Maybe. Language model learning exploits a much richer task compared to the categories in typical annotations
  - Of course, we still fine tune, test, etc.
Final thoughts

• Is linguistic structure all a mistake?
  • No! Deep contextual word representations have phase-shifted from statistical association learners to language discovery devices!
  • Syntax, coref, etc. emerges (approximately) in the geometry of BERT! See:
    • Kevin Clark, Urvashi Khandelwal, Omer Levy, & Christopher Manning. 2019. What Does BERT Look At? An Analysis of BERT’s Attention. BlackBoxNLP.

• Does going big stretch any analogy to child language acquisition?
  • Maybe, but it’s more that acquisition without grounding is unrealistic
Deep Contextual Neural Word Representations: Linguistic Structure Discovery and Efficient Discriminative Training

Christopher Manning
Stanford University and CIFAR Fellow
@chrmanning ⬆️ @stanfordnlp

ElementAI/MILA, December 2019 (last talk of 2019!)