Transfer Learning with NLP Tasks

Irene Li
March 30th, 2020
Motivation
Motivation: performance drop
Motivation: lack of training data

Well-trained Models
NEWS ONLY

Transfer to other domains...

Amazon

Twitter: Cristina

On days like these I would rather work from the car! Gotta muster up the run to the office! Happy...
Cases in NLP? Different Distributions...

<table>
<thead>
<tr>
<th>Electronics</th>
<th>Video games</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compact</strong>; easy to operate; very good picture quality; looks <strong>sharp</strong>!</td>
<td>(2) A very good game! It is action packed and full of excitement. I am very much <strong>hooked</strong> on this game.</td>
</tr>
<tr>
<td>(3) I purchased this unit from Circuit City and I was very <strong>excited</strong> about the quality of the picture. It is really nice and <strong>sharp</strong>.</td>
<td>(4) Very realistic shooting action and good plots. We played this and were <strong>hooked</strong>.</td>
</tr>
<tr>
<td><strong>X</strong></td>
<td>(5) It is also quite <strong>blurry</strong> in very dark settings. I will <strong>never</strong> buy HP again.</td>
</tr>
<tr>
<td>(6) It is so boring. I am extremely unhappy and will probably <strong>never</strong> buy UbiSoft again.</td>
<td></td>
</tr>
</tbody>
</table>
Cases in NLP? Different Distributions...

<table>
<thead>
<tr>
<th>Electronics</th>
<th>Video games</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Compact; easy to operate; very good picture quality; looks sharp!</td>
<td>(2) A very good game! It is action packed and full of excitement. I am very much hooked on this game.</td>
</tr>
<tr>
<td>(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and sharp.</td>
<td>(4) Very realistic shooting action and good plots. We played this and were hooked.</td>
</tr>
<tr>
<td>(5) It is also quite blurry in very dark settings. I will never buy HP again.</td>
<td>(6) It is so boring. I am extremely unhappy and will probably never buy UbiSoft again.</td>
</tr>
</tbody>
</table>

- Source specific: compact, sharp, blurry.
- Target specific: hooked, realistic, boring.
- Domain independent: good, excited, nice, never_buy, unhappy.
Transfer Learning: overview

Inductive TL:
\[ D_s, D_t; T_s \neq T_t \]
Multi-task learning: same dataset, different tasks

Transductive TL
\[ D_s \neq D_t; T_s = T_t \]
Domain adaptation: how different are features?

Unsupervised TL
\[ T_s \neq T_t, Y_s, Y_t? \]
Can not learn from the source domain.

Transfer Learning: overview

<table>
<thead>
<tr>
<th>Source Data (not directly related to the task)</th>
<th>labelled</th>
<th>unlabeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Data labelled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fine-tuning</td>
<td></td>
<td>Self-taught Learning</td>
</tr>
<tr>
<td>Multitask Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain Adaptation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target Data unlabeled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero-shot learning</td>
<td></td>
<td>Self-taught Clustering</td>
</tr>
</tbody>
</table>

Transfer Learning Tutorial (Hung-yi Lee) 2016
Multi-task Learning

Share layers, change task-specific layers

Multi-Task Learning for Multiple Language Translation
Multi-task Learning

Multi-Task Learning for Multiple Language Translation
Zero-shot learning

Unseen training samples

- Same distribution:
  Share similar features!

- Different tasks

In my training dataset:
Chimp, Dog

http://speech.ee.ntu.edu.tw/~tlkagk/

In my testing dataset:
Fish
A toy example...

Training

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

furry 4 legs tail

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```

furry 4 legs tail

```
```

class

```
```

Database

attributes

<table>
<thead>
<tr>
<th>furry</th>
<th>4 legs</th>
<th>tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Fish</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Chimp</td>
<td>O</td>
<td>X</td>
</tr>
</tbody>
</table>

... sufficient attributes for one to one mapping

http://speech.ee.ntu.edu.tw/~tlkagk/
A toy example...

Find the class with the most similar attributes

<table>
<thead>
<tr>
<th>attributes</th>
<th>furry</th>
<th>4 legs</th>
<th>tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Fish</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Chimp</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

sufficient attributes for one to one mapping

http://speech.ee.ntu.edu.tw/~tlkagk/
Word Embeddings
Feature-based Method -- word embeddings

**Task:** sentiment classification (pos or neg)

**Motivation:** reviews or healthcare domain sentiment classifier?

I’ve been **clean** for about 7 months but even now I still feel like maybe I won’t make it….

I feel like I am getting my life back...

Samples from A-CHESS dataset: a study involving users with alcohol addiction.

**Method:** improve the word embeddings

**Domain Adapted (DA) embeddings**

[Domain Adapted Word Embeddings for Improved Sentiment Classification ACL 2018 (short)](#)
Why word embeddings?

Word2vec and GloVe are old news!

**Generic (G) embeddings:**
- word2vec, GloVe trained from Wikipedia, WWW;
- general knowledge

**Domain Specific (DS) embeddings:**
- trained from domain datasets (small-scale);
- domain knowledge

**Domain Adapted (DA) embeddings:** combine them!
Why word embeddings?

Domain-Adapted Embeddings:
- Canonical Correlation Analysis (CCA)
- Kernel CCA (KCCA, nonlinear version of CCA, using RBF kernel)

Word embeddings to sentence encoding:
  i.e. a weighted combination of their constituent word embeddings.

Use a Logistic Regressor to do classification (pos or neg).

Domain Adapted Word Embeddings for Improved Sentiment Classification ACL 2018 (short)
Intuition: Combine two embedding feature space

\[ \bar{w}_{i,DS} = w_{i,DS} \phi_{DS} \]
\[ \bar{w}_{i,G} = w_{i,G} \phi_{G}. \]

CCA maximizes the correlation between \( \bar{w}_{i,DS} \) and \( \bar{w}_{i,G} \) to obtain \( \phi_{DS} \) and \( \phi_{G} \) such that

\[ \rho(\phi_{DS}, \phi_{G}) = \max_{\phi_{DS}, \phi_{G}} \frac{E[\langle \bar{w}_{i,DS}, \bar{w}_{i,G} \rangle]}{\sqrt{E[\bar{w}_{i,DS}^2]E[\bar{w}_{i,G}^2]}} \]  \( \text{(2)} \)

Find out

\[ \phi_{DS} \]
\[ \phi_{G} \]
Canonical Correlation Analysis (CCA)

**Canonical Correlation Analysis (CCA):** \( X = (X_1, \ldots, X_n) \) and \( Y = (Y_1, \ldots, Y_m) \) of random variables, and there are correlations among the variables, then canonical-correlation analysis will find linear combinations of \( X \) and \( Y \) which have maximum correlation with each other.

\[
\tilde{w}_{i,DS} = \begin{bmatrix} w_{i,DS} \\ \phi_{DS} \end{bmatrix} \\
\tilde{w}_{i,G} = \begin{bmatrix} w_{i,G} \\ \phi_{G} \end{bmatrix}.
\]

**Domain-Adapted Embedding:**

\[
\min_{\alpha,\beta} \left\| \tilde{w}_{i,DS} - (\alpha \tilde{w}_{i,DS} + \beta \tilde{w}_{i,G}) \right\|^2 + \left\| \tilde{w}_{i,G} - (\alpha \tilde{w}_{i,DS} + \beta \tilde{w}_{i,G}) \right\|^2.
\]

\[
\hat{w}_{i,DA} = \frac{1}{2} \tilde{w}_{i,DS} + \frac{1}{2} \tilde{w}_{i,G}
\]

**Final Embedding**
### Result on Yelp Dataset

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Embedding</th>
<th>Avg Precision</th>
<th>Avg F-score</th>
<th>Avg AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>W\textsubscript{DA}</td>
<td>KCCA(Glv, LSA)</td>
<td>85.36±2.8</td>
<td>81.89±2.8</td>
<td>82.57±1.3</td>
</tr>
<tr>
<td></td>
<td>CCA(Glv, LSA)</td>
<td>83.69±4.7</td>
<td>79.48±2.4</td>
<td>80.33±2.9</td>
</tr>
<tr>
<td></td>
<td>KCCA(w2v, LSA)</td>
<td>87.45±1.2</td>
<td>83.36±1.2</td>
<td>84.10±0.9</td>
</tr>
<tr>
<td></td>
<td>CCA(w2v, LSA)</td>
<td>84.52±2.3</td>
<td>80.02±2.6</td>
<td>81.04±2.1</td>
</tr>
<tr>
<td></td>
<td>KCCA(GlvCC, LSA)</td>
<td>88.11±3.0</td>
<td>85.35±2.7</td>
<td>85.80±2.4</td>
</tr>
<tr>
<td></td>
<td>CCA(GlvCC, LSA)</td>
<td>83.69±3.5</td>
<td>78.99±4.2</td>
<td>80.03±3.7</td>
</tr>
<tr>
<td></td>
<td>KCCA(w2v, DSw2v)</td>
<td>78.09±1.7</td>
<td>76.04±1.7</td>
<td>76.66±1.5</td>
</tr>
<tr>
<td></td>
<td>CCA(w2v, DSw2v)</td>
<td>86.22±3.5</td>
<td>84.35±2.4</td>
<td>84.65±2.2</td>
</tr>
<tr>
<td></td>
<td>concSVD(Glv, LSA)</td>
<td>80.14±2.6</td>
<td>78.50±3.0</td>
<td>78.92±2.7</td>
</tr>
<tr>
<td></td>
<td>concSVD(w2v, LSA)</td>
<td>85.11±2.3</td>
<td>83.51±2.2</td>
<td>83.80±2.0</td>
</tr>
<tr>
<td></td>
<td>concSVD(GlvCC, LSA)</td>
<td>84.20±3.7</td>
<td>80.39±3.7</td>
<td>80.83±3.9</td>
</tr>
<tr>
<td>W\textsubscript{G}</td>
<td>GloVe</td>
<td>77.13±4.2</td>
<td>72.32±7.9</td>
<td>74.17±5.0</td>
</tr>
<tr>
<td></td>
<td>GloVe-CC</td>
<td>82.10±3.5</td>
<td>76.74±3.4</td>
<td>78.17±2.7</td>
</tr>
<tr>
<td></td>
<td>word2vec</td>
<td>82.80±3.5</td>
<td>78.28±3.5</td>
<td>79.35±3.1</td>
</tr>
<tr>
<td>W\textsubscript{DS}</td>
<td>LSA</td>
<td>75.36±5.4</td>
<td>71.17±4.3</td>
<td>72.57±4.3</td>
</tr>
<tr>
<td></td>
<td>word2vec</td>
<td>73.08±2.2</td>
<td>70.97±2.4</td>
<td>71.76±2.1</td>
</tr>
</tbody>
</table>

**Yelp:** 1000 balanced reviews

**Tokens:** 2049

---

Domain Adapted Word Embeddings for Improved Sentiment Classification ACL 2018 (short)
Feature-based Method: share word embeddings

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Embedding</th>
<th>Avg Precision</th>
<th>Avg F-score</th>
<th>Avg AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-CHESS</td>
<td>KCCA(Glv, LSA)</td>
<td>32.07±1.3</td>
<td>39.32±2.5</td>
<td>65.96±1.3</td>
</tr>
<tr>
<td>DA</td>
<td>CCA(Glv, LSA)</td>
<td>32.70±1.5</td>
<td>35.48±4.2</td>
<td>62.15±2.9</td>
</tr>
<tr>
<td></td>
<td>KCCA(w2v, LSA)</td>
<td>33.45±1.3</td>
<td>39.81±1.0</td>
<td>65.92±0.6</td>
</tr>
<tr>
<td></td>
<td>CCA(w2v, LSA)</td>
<td>33.06±3.2</td>
<td>34.02±1.1</td>
<td>60.91±0.9</td>
</tr>
<tr>
<td></td>
<td>KCCA(GlvCC, LSA)</td>
<td>36.38±1.2</td>
<td>34.71±4.8</td>
<td>61.36±2.6</td>
</tr>
<tr>
<td></td>
<td>CCA(GlvCC, LSA)</td>
<td>32.11±2.9</td>
<td>36.85±4.4</td>
<td>62.99±3.1</td>
</tr>
<tr>
<td></td>
<td>KCCA(w2v, DSw2v)</td>
<td>25.59±1.2</td>
<td>28.27±3.1</td>
<td>57.25±1.7</td>
</tr>
<tr>
<td></td>
<td>CCA(w2v, DSw2v)</td>
<td>24.88±1.4</td>
<td>29.17±3.1</td>
<td>57.76±2.0</td>
</tr>
<tr>
<td></td>
<td>concSVD(Glv, LSA)</td>
<td>27.27±2.9</td>
<td>34.45±3.0</td>
<td>61.59±2.3</td>
</tr>
<tr>
<td></td>
<td>concSVD(w2v, LSA)</td>
<td>29.84±2.3</td>
<td>36.32±3.3</td>
<td>62.94±1.1</td>
</tr>
<tr>
<td></td>
<td>concSVD(GlvCC, LSA)</td>
<td>28.09±1.9</td>
<td>35.06±1.4</td>
<td>62.13±2.6</td>
</tr>
<tr>
<td>W_G</td>
<td>GloVe</td>
<td>30.82±2.0</td>
<td>33.67±3.4</td>
<td>60.80±2.3</td>
</tr>
<tr>
<td></td>
<td>GloVe-CC</td>
<td><strong>38.13±0.8</strong></td>
<td>27.45±3.1</td>
<td>57.49±1.2</td>
</tr>
<tr>
<td></td>
<td>word2vec</td>
<td>32.67±2.9</td>
<td>31.72±1.6</td>
<td>59.64±0.5</td>
</tr>
<tr>
<td>W_DS</td>
<td>LSA</td>
<td>27.42±1.6</td>
<td>34.38±2.3</td>
<td>61.56±1.9</td>
</tr>
<tr>
<td></td>
<td>word2vec</td>
<td>24.48±0.8</td>
<td>27.97±3.7</td>
<td>57.08±2.5</td>
</tr>
</tbody>
</table>

Result on A-CHESS Dataset

A_CHESS: 8% unbalanced

Total: 2500 samples

Tokens: 3400

Domain Adapted Word Embeddings for Improved Sentiment Classification ACL 2018 (short)
About this method...

Straightforward: modify the word embeddings;
Easy to implement;
Possible to improve on sentence embeddings as well.
A typical type before powerful embeddings came out.

Small datasets (thousands for training and testing).
Improvements on classification, what about other tasks?
Pre-BERT (before 2018 Oct.)
CNN Features: pre-train a VGG-19?

Lower level features are shared..

http://cs231n.github.io/convolutional-networks/
Universal Language Model Fine-tuning for Text Classification

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io
Introduction

**Motivation:** Pre-train a VGG-19 in NLP? “Transferring knowledge”

**Challenges:**

- Existing research: train from scratch.
- Small datasets & forgetting: overfitting when fine-tuning.

**Contribution:**

- Propose ULMFit model for any task with robustness
- 3 techniques in fine-tuning: discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing
LM Pre-training

Wikitext-103 dataset

28,595 articles, 103m words

Run once.

Predict next word.
LM Fine-tuning (1)

On target task datasets:

diff dist, small ds, quick

Discriminative fine-tuning

for layers

\[ \theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^l} J(\theta) \]

\[ \eta^{l-1} = \eta^l / 2.6 \]
LM Fine-tuning(2)

Slanted triangular learning rates (through iterations)

Converged quickly.

linearly increase:

learn fast!

linearly decreases:

prevent overfitting

Figure 2: The slanted triangular learning rate schedule used for ULMFiT as a function of the number of training iterations.
LM classifier fine-tuning

Gradual unfreezing

"catastrophic forgetting"

unfreeze from the last layer

contains least generalized knowledge
Datasets

Classification

public datasets

Diverse in length, number of classes, scale and domain

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th># classes</th>
<th># examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC-6</td>
<td>Question</td>
<td>6</td>
<td>5.5k</td>
</tr>
<tr>
<td>IMDb</td>
<td>Sentiment</td>
<td>2</td>
<td>25k</td>
</tr>
<tr>
<td>Yelp-bi</td>
<td>Sentiment</td>
<td>2</td>
<td>560k</td>
</tr>
<tr>
<td>Yelp-full</td>
<td>Sentiment</td>
<td>5</td>
<td>650k</td>
</tr>
<tr>
<td>AG</td>
<td>Topic</td>
<td>4</td>
<td>120k</td>
</tr>
<tr>
<td>DBpedia</td>
<td>Topic</td>
<td>14</td>
<td>560k</td>
</tr>
</tbody>
</table>

Table 1: Text classification datasets and tasks with number of classes and training examples.
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Test</th>
<th>Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoVe (McCann et al., 2017)</td>
<td>8.2</td>
<td>CoVe (McCann et al., 2017)</td>
<td>4.2</td>
</tr>
<tr>
<td>oh-LSTM (Johnson and Zhang, 2016)</td>
<td>5.9</td>
<td>TBCNN (Mou et al., 2015)</td>
<td>4.0</td>
</tr>
<tr>
<td>Virtual (Miyato et al., 2016)</td>
<td>5.9</td>
<td>LSTM-CNN (Zhou et al., 2016)</td>
<td>3.9</td>
</tr>
<tr>
<td>ULMFiT (ours)</td>
<td><strong>4.6</strong></td>
<td>ULMFiT (ours)</td>
<td><strong>3.6</strong></td>
</tr>
</tbody>
</table>

Table 2: Test error rates (%) on two text classification datasets used by McCann et al. (2017).

<table>
<thead>
<tr>
<th>Model</th>
<th>AG</th>
<th>DBpedia</th>
<th>Yelp-bi</th>
<th>Yelp-full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char-level CNN (Zhang et al., 2015)</td>
<td>9.51</td>
<td>1.55</td>
<td>4.88</td>
<td>37.95</td>
</tr>
<tr>
<td>CNN (Johnson and Zhang, 2016)</td>
<td>6.57</td>
<td>0.84</td>
<td>2.90</td>
<td>32.39</td>
</tr>
<tr>
<td>DPCNN (Johnson and Zhang, 2017)</td>
<td>6.87</td>
<td>0.88</td>
<td>2.64</td>
<td>30.58</td>
</tr>
<tr>
<td>ULMFiT (ours)</td>
<td><strong>5.01</strong></td>
<td><strong>0.80</strong></td>
<td><strong>2.16</strong></td>
<td><strong>29.98</strong></td>
</tr>
</tbody>
</table>

Table 3: Test error rates (%) on text classification datasets used by Johnson and Zhang (2017).
Analysis: low-shot learning

**supervised**: only labeled examples are used for LM fine-tuning

**semi-supervised**: all task data is available and can be used to fine-tune the LM

Figure 3: Validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (from left to right).
Analysis: low-shot learning

**supervised**: only labeled examples are used for LM fine-tuning

**semi-supervised**: all task data is available and can be used to fine-tune the LM

Figure 3: Validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (from left to right).
Comments

Novel techniques for fine-tuning; CV-like idea.

Various scale of datasets (lengths, class numbers)

Fine-tuning works well and stable for small datasets: low-resource

Only on classification tasks (model), but potentials on other tasks.

Next: better models to improve pre-trained language modeling, embeddings?
Improve the LMs/embeddings...

Recent breakthroughs:

ELMo: Embeddings from Language Models

OpenAI GPT-2: Language Models are Unsupervised Multi-task Learners

BERT: Bidirectional Transformers for Language Understanding

...
Depending on its context...

Deep contextualized word representations

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†,
{matthewp, markn, mohiti, mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer†*
{csquared, kentonl, lsz}@cs.washington.edu

†Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington
Each token $t_k$

L-layer biLM computes $2L+1$ representations

$k$ is the $k$-th token

$j$ is the $j$-th biLM layer

http://www.realworldnlpbook.com/blog/improving-sentiment-analyzer-using-elmo.html
BERT and its Variations
Finally, a Machine That Can Finish Your Sentence

Completing someone else’s thought is not an easy trick for A.I. But new systems are starting to crack the code of natural language.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin  Ming-Wei Chang  Kenton Lee  Kristina Toutanova
Google AI Language
{JacobDevlin,MingWeichang,KentonL,KristinaT}@google.com
**LM: LSTMs to Transformers**

2017 NIPS

---

**Attention Is All You Need**

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashish Vaswani*</td>
<td>Google Brain</td>
<td><a href="mailto:avaswani@google.com">avaswani@google.com</a></td>
</tr>
<tr>
<td>Noam Shazeer*</td>
<td>Google Brain</td>
<td><a href="mailto:noam@google.com">noam@google.com</a></td>
</tr>
<tr>
<td>Niki Parmar*</td>
<td>Google Research</td>
<td><a href="mailto:nikip@google.com">nikip@google.com</a></td>
</tr>
<tr>
<td>Jakob Uszkoreit*</td>
<td>Google Research</td>
<td><a href="mailto:usz@google.com">usz@google.com</a></td>
</tr>
<tr>
<td>Llion Jones*</td>
<td>Google Research</td>
<td><a href="mailto:llion@google.com">llion@google.com</a></td>
</tr>
<tr>
<td>Aidan N. Gomez* †</td>
<td>University of Toronto</td>
<td><a href="mailto:aidan@cs.toronto.edu">aidan@cs.toronto.edu</a></td>
</tr>
<tr>
<td>Łukasz Kaiser*</td>
<td>Google Brain</td>
<td><a href="mailto:lukaszkaiser@google.com">lukaszkaiser@google.com</a></td>
</tr>
<tr>
<td>Illia Polosukhin* †</td>
<td></td>
<td><a href="mailto:illia.polosukhin@gmail.com">illia.polosukhin@gmail.com</a></td>
</tr>
</tbody>
</table>
BERT Features...

**Bi-directional Transformer** seq2seq model

 Much stronger encoder and decoder.

**Task 1**: Masked Language Modeling

 Replace words randomly: “my dog is cute” -> “my dog is [MASK]”

**Task 2**: Next Sentence Prediction (QA, NLI)

 Binary classification: (sentence1, sentence2) -> IsNext or NotNext?
## Results for QA as an example

<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><strong>Leaderboard (Oct 8th, 2018)</strong></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><strong>Published</strong></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><strong>Ours</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT&lt;sub&gt;BASE&lt;/sub&gt; (Single)</td>
<td>80.8</td>
<td>88.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt; (Single)</td>
<td>84.1</td>
<td>90.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt; (Ensemble)</td>
<td>85.8</td>
<td>91.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt; (Sgl.+TriviaQA)</td>
<td><strong>84.2</strong></td>
<td><strong>91.1</strong></td>
<td><strong>85.1</strong></td>
<td><strong>91.8</strong></td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt; (Ens.+TriviaQA)</td>
<td><strong>86.2</strong></td>
<td><strong>92.2</strong></td>
<td><strong>87.4</strong></td>
<td><strong>93.2</strong></td>
</tr>
</tbody>
</table>

Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.
BERT variations

**RoBERTa**: improving BERT by the training scale (more data, larger batch size), keep the basic BERT main idea

**XLNet**: improving BERT by changing the task -> Permutation Language Modeling, new changes in positional encodings

**ALBERT**: improving computation efficiency of BERT (a lite BERT) -> share parameters across layers; add a task “Sentence Order Prediction”

**DistilBERT, DistilGPT2**: reduce the size of a BERT model by 40%

**BART**: a seq2seq BERT model for Natural Language generation, translation, and comprehension
Cross-lingual BERT

XLM: Cross-lingual Language Model Pretraining (Facebook)

mBERT: multi-lingual BERT

mBART: Multilingual Denoising Pre-training for Neural Machine Translation (Facebook)

XLM-RoBERTa: multi-lingual RoBERTa

More in https://github.com/huggingface/transformers (Model architectures)
Understanding BERT
BERTology: paper 40 analysis studies

Motivation: why? Less cognitive motivation

What knowledge does BERT have?

- fill-in-the-gap probes of BERT’s MLM,
- analysis of self-attention weights,
- probing classifiers using different BERT representations as inputs.
BERT embeddings

Syntactic Knowledge

BERT representations are hierarchical rather than linear: syntactic tree structure encode information about parts of speech, syntactic chunks and roles syntactic structure can be extracted indirectly: Hewitt and Manning (2019)

Semantic Knowledge

encodes information about entity types, relations, semantic roles, and proto-roles; struggles with representations of numbers;
Localizing linguistic knowledge

Syntactic information is the most prominent in the middle BERT layers (8-9)

subject-verb agreement

Final layers are the most task-specific;

Semantics is spread across the entire model of all layers

.. all token level, what about sentences and paragraphs?
A Structural Probe for Finding Syntax in Word Representations

We’ll present a method for finding tree structures in these vector spaces, and show the surprising extent to which ELMo and BERT encode human-like parse trees.

They showed evidence that entire syntax trees are embedded implicitly in deep models’ vector geometry.

An example: *The chef who ran to the store was out of food.*
A Structural Probe for Finding Syntax in Word Representations

**post**

Find a linear mapping, connect to the closest word...approximates the human parse tree! → structural hypothesis

equivalently, it is finding the distance on the original space that best fits the tree metrics
The structural probe

**Intuition:** vector spaces and graphs both have natural distance metrics.

Path metric: how many steps...

What about…vector distances!

**The syntax distance hypothesis:** There exists a linear transformation $B$ of the word representation space under which vector distance encodes parse trees.

Equivalently, there exists an inner product on the word representation space such that distance under the inner product encodes parse trees. This (indefinite) inner product is specified by $B^T B$. 

Properties

**Distance**: symmetric (A-B) vs (B-A)

Implies proximity not a word is governed by another.

**Tree Depth**: directions of the edges in a parse tree is determined by the depth of words in the parse tree

The number of edges in the parse tree between and the root and wi

\[
\min_B \sum_{\ell} \frac{1}{s_\ell} \sum_i (\|w_i\| - \|Bh_i\|^2)
\]
Experiments

Models: ELMo[\text{dim 1024}];

- Bert-base (cased) [\text{dim 768}];
- and Bert-large (cased) [\text{dim 1024}]

Data: \textit{Stanford Dependencies formalism} in English and Penn Treebank (training)

\textbf{Baselines} are non-contextualized embeddings, linear, etc. (Board)
Evaluation

**Tree distance evaluation**: look at all word pairs
   - Compare minimum spanning tree of the test sentence parse trees.
   - UUAS: undirected unlabeled attachment score—>the percent of undirected edges placed correctly—against the gold tree.
   - Spearman distance: a squared Euclidean distance; sentence lengths 5–50

**Tree depth evaluation**: order of words specified by their depth.
   - Also looked at the root node.
Results

Figure 2: Minimum spanning trees resultant from predicted squared distances on BERTLARGE16 and ELM01 compared to the best baseline, PROJ0. Black edges are the gold parse, above each sentence; blue are BERTLARGE16, red are ELM01, and purple are PROJ0.
Conclusion and Takeaway

**Syntactic information** is the most prominent in the middle BERT layers (8-9)

subject-verb agreement

In contextualized embeddings, there are richer information: shows a global structural property.

Why only uses a single linear transformation? Because the space is dense and rich.

Map entities into the high-dimension space, reconstruct based on L2 distance. May be useful for testing the existence of different types of graph structures. Entity linking, what else?
Knowledge Distillation: Reducing BERT
Knowledge Distillation

Use a function to approximate the current model!

\[
\min_{\theta_f} \text{Dist}(y_b, y_f)
\]
DistillBERT by Hugging Face (post)

Reducing the size:
- Quantization: approximating the weights of a network with a smaller precision
- Weights pruning: removing some connections in the network
- Distillation: compress a large model (teacher) to a smaller model (student)

Transferring generalization capabilities:
- Prevents the model to be too sure about its prediction;
- Dark knowledge: long tail labels

How to solve? One-hot encoding for gold labels changed to cross-entropy, (right)
softmax-temperature

\[ L = - \sum_i t_i * \log(s_i) \]

\[ p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \]

Training loss: distillation loss (KL loss between model parameters) and the
masked language modeling loss.

Results: Half size of BERT; 95% performance!
Electronic health records: Transfer learning in real world
Quick survey: transfer learning + EHR

RNN, LSTM research on time-series data: [We don’t want to work on this for now]

- **structured data (mainly)**, predict events, death, etc.
- pre-training, fine-tuning to transfer

With Text data for use cases:

- Various apps/use cases: Adverse Event Reporting (NER), event prediction/recognition, Mortality Prediction, etc.
- Deep-based methods: CNNs, RNNs, BERT, word embeddings.
- Adapting a model to a local setting for personalized data.

Related to Text data:

- Embedding learning: ICD-9 code, patients, etc.
Use case: Using EHR Clinical Notes to Predict Death

Data source could be MIMIC-III.

We can apply better models using transfer learning. Heart failure/mortality prediction?

**FIGURE 8.** Use case scenario clinical notes are vectorized using Word2Vec skip gram model, then using labels obtained from patient history weather he died or not, train a CNN model which can predict a near future death prediction using patient's hospital notes.
Transfer Learning for Biomedical Named Entity Recognition with BioBERT

- Data: https://github.com/AnthiS/MasterThesis_DS/tree/master/biobert_ner
- Relevant paper: BioBERT

Transfer Learning for Named-Entity Recognition with Neural Networks

- MIMIC and i2b2 datasets.
- Code: https://github.com/Franck-Demoncourt/NeuroNER
- Study the impact of transfer learning (transfer a model trained on a large dataset to a small one)

Using Transfer Learning for Improved Mortality Prediction in a Data-Scarce Hospital Setting

- MIMIC data, structured data, decision tree
- Can we apply same pipeline but using text data?

Recent review:
- Deep Learning for Electronic Health Records Analytics
Wrapup: we have covered...

Transfer Learning with word embeddings

Pre-BERT times

BERT golden time

Understanding, reducing BERT

Transfer Learning in real world
Discussion

Transfer Learning works in graph-structured data:
   Are there any?

Recent efforts:
   General domain datasets: in ‘domains’-> ‘News’ vs ‘Tweets’; ‘General’ vs ‘Medical’, etc
   Muli-language datasets: large on scale and languages!
      A single normal GPU is very hard to handle them!
   BERT-based Models: are the improvements brought by BERT?
THANKS!

Q&A

ireneli.eu