Generalization
(or lack thereof)

Changyan Wang and Ben Dodge
Learning and Evaluating General Linguistic Intelligence

Dani Yogatama, Cyprien de Masson d’Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, Phil Blunsom
DeepMind, London, United Kingdom
{dyogatama, cyprien, jeromeconnor, tkocisky, chrzanowskim}@google.com
{lingpengk, angeliki, lingwang, leiyu, cdyer, pblunsom}@google.com

BAM! Born-Again Multi-Task Networks for Natural Language Understanding

Kevin Clark†  Minh-Thang Luong‡  Urvashi Khandelwal†
Christopher D. Manning†  Quoc V. Le‡
†Computer Science Department, Stanford University
‡Google Brain
{kevclark, urvashik, manning}@cs.stanford.edu
{thangluong, qvl}@google.com
What is general linguistic intelligence?

Criteria from the paper:

(i) deal with the full complexity of natural language across a variety of tasks

(ii) effectively store and reuse representations, combinatorial modules, and previously acquired linguistic knowledge to avoid catastrophic forgetting

(iii) adapt to new linguistic tasks in new environments with little experience

And why do we care?
Where are we now?

Datasets/metrics do not focus on *generalization* or *abstraction*.

Models are not evaluated on all datasets from a given task.

**Code length evaluation**

- **MRQA 2019**

**BAM!**

BERT

- Unsupervised pretraining enables transfer to many tasks

- Multi-task training provides a pathway to general intelligence
Paper Outline

➢ New evaluation metric
➢ Tasks & datasets
➢ Models
➢ Five interesting questions
New evaluation metric

**Codelength** aims to measure the number of task-specific training examples needed to reach high performance.

\[ \ell(\mathcal{A}) = \log_2 |y| - \sum_{i=2}^{N} \log_2 p(y_i | x_i; \hat{\mathbf{W}}_{\mathcal{A}_{i-1}}) \]

- Dataset
- Number of classes
- Parameters trained on examples 1 through \(i-1\)
Interpretation (Blier & Ollivier, 2018)

Alice has all \((x, y)\) pairs and Bob only has the \(x\). Alice wants to send \(y\) to Bob.

**Proposition 1** (Shannon–Huffman code). Suppose that Alice and Bob have agreed in advance on a model \(p\), and both know the inputs \(x_{1:n}\). Then there exists a code to transmit the labels \(y_{1:n}\) losslessly with codelength (up to at most one bit on the whole sequence)

\[
L_p(y_{1:n}|x_{1:n}) = - \sum_{i=1}^{n} \log_2 p(y_i|x_i)
\]

(2.1)

We are not concerned with how to do this in practice.
Don’t use any deep nets at all, use a uniform model \( (K \text{ is number of classes}) \)

\[
p(y_i | x_i) = \frac{1}{K} \quad \rightarrow \quad \ell(A) = N \log_2 K
\]

This is the same as if Alice just sent all the labels to Bob with no model.
Two-Part Encoding

1. Alice trains a deep net and sends the parameters $\theta$ to Bob
2. Alice uses the deep net to transmit the labels more efficiently

\[ L_p(y_{1:n}|x_{1:n}) = -\sum_{i=1}^{n} \log_2 p(y_i|x_i) \]

\[ \ell(A) = \ell(\theta) - \sum_{i=1}^{N} \log_2 p_{\theta}(y_i|x_i) \]

Bits needed to transmit parameters (too large)
1. Alice sends one label
2. Both Alice and Bob train on the label
3. Alice uses resulting deep net to send the next label

$\ell(A) = \log_2 K - \sum_{i=2}^{N} \log_2 p_{\theta_{A_{i-1}}} (y_i | x_i)$

$L_p(y_{1:n} | x_{1:n}) = - \sum_{i=1}^{n} \log_2 p(y_i | x_i)$
More about code length

- Chaitin’s hypothesis: “comprehension is compression”

- Expensive to compute for every training example, so split into subsets

\[ \ell(A) = |S_1| \log_2 |Y| - \sum_{i=2}^{M} \log_2 p(y_{S_i} \mid x_{S_i}, \hat{W}_{S_{i-1}}) \]

- How to do for span selection tasks?
Main Tasks

**READING COMPREHENSION**

★ **SQuAD 1.1**
  - questions constructed from Wikipedia passages
  - 90k train / 10k val

**TriviaQA**
  - trivia questions & answers, evidence from the web
  - 76k train / 300 val

**QuAC**
  - information-seeking dialogue, response spans from Wikipedia
  - 80k train / 7k val

**NATURAL LANGUAGE INFERENCE**

★ **MNLI**
  - multi-genre entailment
  - 400k train / 20k test

**SNLI**
  - 550k train / 10k test
TriviaQA (Joshi et al. 2017)

• QA pairs collected from 14 trivia websites
• Evidence filtered from Bing, Wikipedia
• Only documents which contain answer
• Multiple training examples per QA pair

Question: The Dodecanese Campaign of WWII that was an attempt by the Allied forces to capture islands in the Aegean Sea was the inspiration for which acclaimed 1961 commando film?
Answer: The Guns of Navarone
Excerpt: The Dodecanese Campaign of World War II was an attempt by Allied forces to capture the Italian-held Dodecanese islands in the Aegean Sea following the surrender of Italy in September 1943, and use them as bases against the German-controlled Balkans. The failed campaign, and in particular the Battle of Leros, inspired the 1957 novel The Guns of Navarone and the successful 1961 movie of the same name.
QuAC (Choi et al. 2018)

Figure 1: An example dialog about a Wikipedia section. The student, who does not see the section text, asks questions. The teacher provides a response in the form of a text span (or \textcolor{blue}{No answer}), optionally yes or no (\textcolor{red}{Yes} / \textcolor{green}{No}), and encouragement about continuing a line of questioning (should, \textcolor{red}{\leftarrow}, could \textcolor{green}{\rightarrow}, or should not \textcolor{green}{\leftrightarrow} ask a follow-up question).

- Entire context as evidence
- Pros / cons of collection method?
Other Tasks

SEMANTIC ROLE LABELING

FitzGerald et al. (2018)
- SRL as span-prediction
- 200k train / 25k test

RELATION EXTRACTION

Levy et al. (2017)
- slot-filling as Q&A
- 900k train / 5k test

In 1950 Alan M. Turing published "Computing machinery and intelligence" in Mind, in which he proposed that machines could be tested for intelligence using questions and answers.

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>published</td>
<td>Who published something?</td>
<td>Alan M. Turing</td>
</tr>
<tr>
<td></td>
<td>What was published?</td>
<td>&quot;Computing Machinery and Intelligence&quot;</td>
</tr>
<tr>
<td></td>
<td>When was something published?</td>
<td>In 1950</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
<th>Question Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>educated_at(x, y)</td>
<td>Where did x graduate from? In which university did x study? What is x’s alma mater?</td>
</tr>
<tr>
<td>occupation(x, y)</td>
<td>What did x do for a living? What is x’s job? What is the profession of x?</td>
</tr>
<tr>
<td>spouse(x, y)</td>
<td>Who is x’s spouse? Who did x marry? Who is x married to?</td>
</tr>
</tbody>
</table>
Models

**Transformer**

- BERT\textsubscript{BASE}
  - default vocabulary
  - 110M parameters

**RNN**

- ELMo + LSTM + BiDAF
  - character-based
  - 100M ELMo parameters
Experiments

1. How much in-domain training data is needed to obtain good performance?
2. Can pretraining on other datasets and tasks improve performance?
3. Do these models generalize to other datasets from the same task?
4. How fast do these models forget their previously acquired linguistic knowledge?
5. How does curriculum affect performance and how do we design this curriculum?
How much **in-domain training data** is needed to obtain good performance?
Models need about 40,000 training examples
## Code lengths?

<table>
<thead>
<tr>
<th>Model</th>
<th>SQuAD 1.1</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>102.42 kbits</td>
<td>89.25 kbits</td>
</tr>
<tr>
<td>ELMo</td>
<td>112.96 kbits</td>
<td>132.17 kbits</td>
</tr>
</tbody>
</table>
Can pretraining on other datasets and tasks improve performance?
Pretrain on all supervised tasks (SRL, RE, MNLI, SNLI, TriviaQA, QuAC), then train on SQuAD.
Do these models **generalize** to other datasets from the same task?
Evaluate best SQuAD model on other tasks.

<table>
<thead>
<tr>
<th></th>
<th>SQuAD</th>
<th>Trivia</th>
<th>QuAC</th>
<th>QA-SRL</th>
<th>QA-ZRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>86.5 (78.5)</td>
<td>35.6 (13.4)</td>
<td>56.2 (43.9)</td>
<td>77.5 (65.0)</td>
<td>55.3 (40.0)</td>
</tr>
<tr>
<td>ELMo</td>
<td>81.8 (72.2)</td>
<td>32.9 (12.6)</td>
<td>45.0 (34.5)</td>
<td>68.7 (52.3)</td>
<td>60.2 (42.0)</td>
</tr>
</tbody>
</table>

Table 2: $F_1$ (exact match) scores of the best BERT and ELMo models trained on SQuAD and evaluated on other question answering datasets.
How fast do these models forget their previously acquired linguistic knowledge?
Train on one dataset at a time ("continual learning").

unsupervised → SQuAD → MNLI

unsupervised → SQuAD → TriviaQA
Train on one dataset at a time ("continual learning").
How does **curriculum** affect performance and how do we design this curriculum?
Train on all datasets at the same time ("random training curriculum" / "mixed curriculum").

<table>
<thead>
<tr>
<th></th>
<th>SQuAD</th>
<th>Trivia</th>
<th>QuAC</th>
<th>QA-SRL</th>
<th>QA-ZRE</th>
<th>MNLI</th>
<th>SNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>85.4</td>
<td>72.5</td>
<td>60.0</td>
<td>85.0</td>
<td>88.2</td>
<td>81.1</td>
<td>88.0</td>
</tr>
<tr>
<td>ELMo</td>
<td>78.3</td>
<td>57.1</td>
<td>54.3</td>
<td>67.3</td>
<td>88.5</td>
<td>69.1</td>
<td>77.9</td>
</tr>
</tbody>
</table>
Key Takeaways

• Current models **solve datasets, not tasks.** They need significant in-domain training data to attain good performance.

• Ability to **rapidly generalize** can and should be evaluated both *across* datasets and *within* datasets (using codelength, for example).

• Poor generalization is partly due to **task-specific components**, so we might look for ways to unify tasks (text-to-text framework, for example).

• Continual training does not work, as **models forget earlier training.** Only mixed training curricula lead to good multi-task models.
BAM! Born-Again Multi-Task Networks for Natural Language Understanding

Kevin Clark†  Minh-Thang Luong†  Urvashi Khandelwal†  Christopher D. Manning†  Quoc V. Le‡
†Computer Science Department, Stanford University
‡Google Brain

{kevclark, urvashik, manning}@cs.stanford.edu
{thangluong, qvl}@google.com
Outline

- Review of Distillation
- Background / Previous Work
- Context
- BAM's General Approach
- Tricks
- Experiments / Results
Review of Distillation

- Train a teacher model
- Replace gold-label with teacher probability predictions
- E.g. from large model (teacher) to small model

Standard Training Objective

\[ \mathcal{L}(\theta) = \sum_{x_i, y_i \in \mathcal{D}} l(y_i, f(x_i, \theta)) \]

Distillation Training Objective

\[ \mathcal{L}(\theta) = \sum_{x_i, y_i \in \mathcal{D}} l(f(x_i, \theta'), f(x_i, \theta)) \]
**Standard Training Objective**

$$\mathcal{L}(\theta) = \sum_{x_i, y_i \in \mathcal{D}} l(y_i, f(x_i, \theta))$$

**Distillation Training Objective**

$$\mathcal{L}(\theta) = \sum_{x_i, y_i \in \mathcal{D}} l(f(x_i, \theta'), f(x_i, \theta))$$
Outline

- Review of Distillation
- Background / Previous Work
- Context
- BAM's General Approach
- Tricks
- Experiments / Results
• Multi-Task NLP Models
  – Design architecture to share only helpful information (Ruder et al. 2019)
  – BAM is orthogonal
• Distillation
  – Distillation is used in NLP from large -> small models (Kim and Rush 2016)
  – **Born-again models** (Furlanello et al. 2018)
    ■ Large -> Large (same size)
  – Distill single-language-pair translation models into a multi-language model (Tan et al 2019)
• Multi-task BERT: **MT-DNN** (Liu et al. 2019)
Multi-Task NLP: Architecture Changes

(a) Fully Shared Model (FS-MTL)

(b) Shared-Private Model (SP-MTL)
Background / Previous Work

- **Born-again network** (Furlanello et al, 2018)
  - Variant of distillation
  - Teacher, student have same architecture
  - Surprisingly, student does better than teacher!

<table>
<thead>
<tr>
<th>Network</th>
<th>Teacher</th>
<th>BAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DenseNet-112-33</td>
<td>18.25</td>
<td>16.95</td>
</tr>
<tr>
<td>DenseNet-90-60</td>
<td>17.69</td>
<td>16.69</td>
</tr>
<tr>
<td>DenseNet-80-80</td>
<td>17.16</td>
<td>16.36</td>
</tr>
<tr>
<td>DenseNet-80-120</td>
<td>16.87</td>
<td>16.00</td>
</tr>
</tbody>
</table>

Test Error on CIFAR-100
Background / Previous Work

- Multi-task BERT: **MT-DNN** (Liu et al. 2019)
- Mixed curriculum

<table>
<thead>
<tr>
<th>Model</th>
<th>GLUE score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Base (Devlin et al., 2019)</td>
<td>78.5</td>
</tr>
<tr>
<td>BERT-Large (Devlin et al., 2019)</td>
<td>80.5</td>
</tr>
<tr>
<td>BERT on STILTs (Phang et al., 2018)</td>
<td>82.0</td>
</tr>
<tr>
<td>MT-DNN (Liu et al., 2019b)</td>
<td>82.2</td>
</tr>
</tbody>
</table>
Outline

- Review of Distillation
- Background / Previous Work
- Context
- BAM's General Approach
- Tricks
- Experiments / Results
Context

- Catastrophic forgetting
- Is single-task fine-tuning necessary? **Mixed curriculum?**
  - Yogatama et al: Mixed curriculum does okay!
  - Performance lags
    - SQuAD: 86.5 -> 85.4
    - MNLI: 84.6 -> 81.1
  - MT-DNN: Mixed curriculum yields **stronger** performance!
- Can mixed curriculum beat fine-tuned models?
  - BAM: Yes, using some tricks
    - Distill many teachers into a single multi-task model
- Note: BAM doesn't resolve continual learning
Outline

- Review of Distillation
- Background / Previous Work
- Context
- BAM's General Approach
- Tricks
- Experiments / Results
BAM: General Approach

- **Multi-task BERT, with single-task teachers**
  - Train many single-task models, use as teachers for multi-task model
- **Main tricks:**
  - Many teachers, one per task
  - Born-again: same architecture
  - Teacher annealing
  - Task sampling
  - Different learning rate per layer
Review: Single-Task BERT

- Pre-train using language modeling
- Fine-tune a different BERT model for each single task
Review: Single-Task BERT

- Pre-train using language modeling
- Fine-tune a different BERT model for each single task
  - Add a new final layer on top of the pre-trained network
  - For classification tasks, use softmax
    - $\text{softmax}(Wc)$
  - For regression tasks, normalize labels and use sigmoid activation
    - $\text{sigmoid}(w^Tc)$
Multi-task BERT in BAM

- Same architecture as standard BERT
- For multi-task model, **only change final layer**
  - All other parameters shared between tasks!
- **Mixed curriculum**
  - Different tasks are mixed
  - Each minibatch contains multiple tasks
- Training objective
  - Either use standard gold-label training
  - Or use distillation (using a BERT teacher)
    - Born-again, since teacher has same architecture as student
    - Clarify: Single vs Multi-task teacher
Single vs. Multi-task BERT

- Pre-training BERT
- Fine-Tuning MNLI
- Fine-Tuning SQuAD
- Fine-Tuning MNLI and SQUAD
BAM vs. MT-DNN

**Shared layers**
- $l_1$: input embedding vectors, one each token.
- Lexicon Encoder (word, position and segment)
- $X$: a sentence or a pair of sentences

**Task specific layers**
- $P_1(c | X)$ (e.g., probability of labeling text $X$ by $c$)
- Sim($X_1$, $X_2$) (e.g., semantic similarity between $X_1$ and $X_2$)
- $P_2(R | P, H)$ (e.g., probability of logic relationship $R$ between $P$ and $H$)
- Rel($Q, A$) (e.g., relevance score of candidate answer $A$ given query $Q$)

**Networks**
- Transformer Encoder (contextual embedding layers)
- BERT

**DNNs**
- MNLI Final Layer
- SQuAD Final Layer
Outline

- Review of Distillation
- Background / Previous Work
- Context
- BAM's General Approach
- Tricks
- Experiments / Results
Teacher Annealing

- Standard distillation: Either train on teacher outputs or gold label
- Teacher annealing: Mix teacher outputs and gold label
- Gradually increase lambda to 1 (use gold labels more over time)

\[
l(y_i, f(x_i, \theta)) \quad l(f(x_i, \theta'), f(x_i, \theta))
\]

\[
l(\lambda y_i + (1 - \lambda)f(x_i, \theta'), f(x_i, \theta))
\]
Other Tricks

- Task Sampling (Bowman et al 2018)
  - Sample an example from a task proportionally to the $\frac{3}{4}$ root of the size of dataset for that task (slightly downweight examples from large datasets)
  - $|D_\tau|^{0.75}$

- Layerwise-learning-rate (Howard and Ruder 2018)
  - Different learning rate for each layer: $BASE_{LR} \times \alpha^d$
  - Layers closest to input get lower learning rate
  - $\alpha = 0.9$ for multi-task models
Outline

- Review of Distillation
- Background / Previous Work
- Context
- BAM's General Approach
- Tricks
- Experiments / Results
Experimental Setup

- Evaluate on GLUE
  - Collection of tasks including question answering, sentiment analysis, and textual entailment
- Compare various versions of BERT
  - **Single** (standard BERT, single-task fine-tuning)
  - **Multi** (mixed curriculum, gold labels)
  - **Single -> Single** (standard BERT, single-task fine-tuning, teachers are single-task learners)
  - **Single -> Multi** (mixed curriculum, teachers are single-task learners)
  - **Multi -> Multi** (mixed curriculum, teachers are multi-task)
  - **Single -> Multi -> Single -> Multi** (multiple rounds of distillation)
Single -> Multi in BAM

- Pre-training
  - BERT

- Fine-Tuning MNLI
  - BERT
  - MNLI Final Layer

- Fine-Tuning SQuAD
  - BERT
  - SQuAD Final Layer

- Fine-Tuning MNLI and SQUAD
  - BERT
  - MNLI Final Layer
  - MNLI Final Layer
Review: GLUE

- Single-sentence tasks
  - CoLA (Is this sentence grammatical?)
  - SST-2 (Sentiment analysis: Is this sentence positive or negative?)
- Similarity and Paraphrase Tasks
  - MRPC, QQP, STS-B (Are these sentences semantically equivalent?)
- Inference Tasks
  - MNLI, QNLI, RTE, WNLI
  - (What is the relationship between these sentences? Entailment, contradiction, or neutral?)
## Results

| Model          | Avg. | CoLA\(^a\)\(|D| = 8.5k\) | SST-2\(^b\) | MRPC\(^c\) | STS-B\(^d\) | QQP\(^e\)  | MNLI\(^f\) | QNLI\(^g\) | RTE\(^h\) |
|----------------|------|-----------------------------|-------------|------------|-------------|------------|-----------|-----------|----------|
| Single         | 84.0 | 60.6                        | 93.2        | 88.0       | 90.0        | 91.3       | 86.6      | 92.3      | 70.4     |
| Multi          | 85.5 | 60.3                        | 93.3        | 88.0       | 89.8        | 91.4       | 86.5      | 92.2      | 82.1     |
| Single→Single  | 84.3 | **61.7**\(^{**}\)           | 93.2        | **88.7**\(^{*}\) | 90.0        | 91.4       | **86.8**\(^{**}\) | 92.5\(^{***}\) | 70.0     |
| Multi→Multi    | 85.6 | 60.9                        | 93.5        | 88.1       | 89.8        | **91.5**\(^{*}\) | 86.7      | 92.3      | 82.0     |
| Single→Multi   | **86.0**\(^{***}\) | **61.8**\(^{**}\)           | **93.6**\(^{*}\) | **89.3**\(^{**}\) | 89.7        | **91.6**\(^{*}\) | **87.0**\(^{***}\) | 92.5\(^{***}\) | **82.8**\(^{*}\) |

Dataset references:
- \(^a\)Warstadt et al. (2018)
- \(^b\)Socher et al. (2013)
- \(^c\)Dolan and Brockett (2005)
- \(^d\)Cer et al. (2017)
- \(^e\)Iyer et al. (2017)
- \(^f\)Williams et al. (2018)
- \(^g\)constructed from SQuAD (Rajpurkar et al., 2016)
- \(^h\)Giampiccolo et al. (2007)

Table 1: Comparison of methods on the GLUE dev set. \(^{*}\), \(^{**}\), and \(^{***}\) indicate statistically significant (\(p < .05\), \(p < .01\), and \(p < .001\)) improvements over both Single and Multi according to bootstrap hypothesis tests.\(^{3}\)
<table>
<thead>
<tr>
<th>Model</th>
<th>Avg.</th>
<th>CoLA</th>
<th>SST-2</th>
<th>MRPC</th>
<th>STS-B</th>
<th>QQP</th>
<th>MNLI</th>
<th>QNLI</th>
<th>RTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>84.0</td>
<td>60.6</td>
<td>93.2</td>
<td>88.0</td>
<td>90.0</td>
<td>91.3</td>
<td>86.6</td>
<td>92.3</td>
<td>70.4</td>
</tr>
<tr>
<td>Multi</td>
<td>85.5</td>
<td>60.3</td>
<td>93.3</td>
<td>88.0</td>
<td>89.8</td>
<td>91.4</td>
<td>86.5</td>
<td>92.2</td>
<td>82.1</td>
</tr>
<tr>
<td>Single→Single</td>
<td>84.3</td>
<td><strong>61.7</strong></td>
<td>93.2</td>
<td><strong>88.7</strong></td>
<td>90.0</td>
<td>91.4</td>
<td>86.8**</td>
<td>92.5***</td>
<td>70.0</td>
</tr>
<tr>
<td>Multi→Multi</td>
<td>85.6</td>
<td>60.9</td>
<td>93.5</td>
<td>88.1</td>
<td>89.8</td>
<td><strong>91.5</strong></td>
<td>86.7</td>
<td>92.3</td>
<td>82.0</td>
</tr>
<tr>
<td>Single→Multi</td>
<td><strong>86.0</strong>*</td>
<td><strong>61.8</strong></td>
<td><strong>93.6</strong></td>
<td><strong>89.3</strong></td>
<td>89.7</td>
<td>91.6*</td>
<td><strong>87.0</strong>*</td>
<td><strong>92.5</strong>*</td>
<td><strong>82.8</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trained Tasks</th>
<th>RTE score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE</td>
<td>70.0</td>
</tr>
<tr>
<td>RTE + MNLI</td>
<td>83.4</td>
</tr>
<tr>
<td>RTE + QQP + CoLA + SST</td>
<td>75.1</td>
</tr>
<tr>
<td>All GLUE</td>
<td>82.8</td>
</tr>
</tbody>
</table>

Table 5: Which tasks help RTE? Pairwise differences are statistically significant ($p < .01$) according to Mann-Whitney U tests. 

- $|D| = 8.5k$ indicates the dataset size of CoLA.
- 67k, 3.7k, and 5.8k represent the dataset sizes of SST-2, MRPC, and STS-B, respectively.
- 364k, 393k, and 108k represent the dataset sizes of QQP, MNLI, and QNLI, respectively.
- 2.5k represents the dataset size of RTE.

Note: 
- ** indicates $p < .01$.
- *** indicates $p < .001$. 
- * indicates $p < .05$.
# Results

<table>
<thead>
<tr>
<th>Model</th>
<th>GLUE score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Base (Devlin et al., 2019)</td>
<td>78.5</td>
</tr>
<tr>
<td>BERT-Large (Devlin et al., 2019)</td>
<td>80.5</td>
</tr>
<tr>
<td>BERT on STILTs (Phang et al., 2018)</td>
<td>82.0</td>
</tr>
<tr>
<td>MT-DNN (Liu et al., 2019b)</td>
<td>82.2</td>
</tr>
<tr>
<td>Span-Extractive BERT on STILTs</td>
<td>82.3</td>
</tr>
<tr>
<td>(Keskar et al., 2019)</td>
<td></td>
</tr>
<tr>
<td>Snorkel MeTaL ensemble (Hancock et al., 2019)</td>
<td>83.2</td>
</tr>
<tr>
<td>MT-DNN$_{KD}$* (Liu et al., 2019a)</td>
<td>83.7</td>
</tr>
<tr>
<td>BERT-Large + BAM (ours)</td>
<td>82.3</td>
</tr>
</tbody>
</table>

Table 2: Comparison of test set results. *MT-DNN$_{KD}$ is distilled from a diverse ensemble of models.
## Ablation

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single→Multi</td>
<td>86.0</td>
</tr>
<tr>
<td>No layer-wise LRs</td>
<td>-0.3</td>
</tr>
<tr>
<td>No task sampling</td>
<td>-0.4</td>
</tr>
<tr>
<td>No teacher annealing: ( \lambda = 0 )</td>
<td>-0.5</td>
</tr>
<tr>
<td>No teacher annealing: ( \lambda = 0.5 )</td>
<td>-0.3</td>
</tr>
</tbody>
</table>

**Table 4:** Ablation Study. Differences from Single→Multi are statistically significant \((p < .001)\) according to Mann-Whitney U tests.³

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>84.0</td>
</tr>
<tr>
<td>Multi</td>
<td>85.5</td>
</tr>
<tr>
<td>Single→Single</td>
<td>84.3</td>
</tr>
<tr>
<td>Multi→Multi</td>
<td>85.6</td>
</tr>
<tr>
<td>Single→Multi</td>
<td><strong>86.0</strong></td>
</tr>
</tbody>
</table>
Conclusion and Caveats

- Multi-task training can perform **better** than single-task training!
- Tricks are important!
  - Teacher annealing, layer-wise learning rate, task sampling
- Single-task fine-tuning isn't necessary?
- BAM doesn't solve continual learning: need mixed curriculum

Criteria from the paper:

(i) deal with the full complexity of natural language across a variety of tasks
(ii) effectively store and reuse representations, combinatorial modules, and previously acquired linguistic knowledge to avoid *catastrophic forgetting*
(iii) adapt to new linguistic tasks in new environments with little experience