GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman

1 New York University, New York, NY
2 Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA
3 DeepMind, London, UK

{alexwang, amanpreet, bowman}@nyu.edu
{julianjm, omerlevy}@cs.washington.edu
felixhill@google.com

Abstract

For natural language understanding (NLU) technology to be maximally useful, both practically and as a scientific object of study, it must be general: it must be able to process language in a way that is not exclusively tailored to any one specific task or dataset. In pursuit of this objective, we introduce the General Language Understanding Evaluation benchmark (GLUE), a tool for evaluating and analyzing the performance of models across a diverse range of existing NLU tasks. GLUE is model-agnostic, but it incentivizes sharing knowledge across tasks because certain tasks have very limited training data. We further provide a hand-crafted diagnostic test suite that enables detailed linguistic analysis of NLU models. We evaluate baselines based on current methods for multi-task and transfer learning and find that they do not immediately give substantial improvements over the aggregate performance of training a separate model per task, indicating room for improvement in developing general and robust NLU systems.

1 Introduction

Human ability to understand language is general, flexible, and robust. We can effectively interpret and respond to utterances of diverse form and function in many different contexts. In contrast, most natural language understanding (NLU) models above the word level are designed for one particular task and struggle with out-of-domain data. If we aspire to develop models whose understanding extends beyond the detection of superficial correspondences between inputs and outputs, then it is critical to understand how a single model can learn to execute a range of different linguistic tasks on language from different domains.

To motivate research in this direction, we present the General Language Understanding Evaluation benchmark (GLUE, gluebenchmark.com), an online tool for evaluating the performance of a single NLU model across multiple tasks, including question answering, sentiment analysis, and textual entailment, built largely on established existing datasets. GLUE does not place any constraints on model architecture beyond the ability to process single-sentence and paired-sentence inputs and to make corresponding predictions. For some GLUE tasks, directly pertinent training data is plentiful, but for others, training data is limited or fails to match the genre of the test set. GLUE therefore favors models that can learn to represent linguistic and semantic knowledge in a way that facilitates sample-efficient learning and effective knowledge transfer across tasks.

Though GLUE is designed to prefer models with general and robust language understanding, we cannot entirely rule out the existence of simple superficial strategies for solving any of the included tasks. We therefore also provide a set of newly constructed evaluation data for the analysis of model performance. Unlike many test sets employed in machine learning research that reflect the frequency distribution of naturally occurring data or annotations, this dataset is designed to highlight points of difficulty that are relevant to model development and training, such as the incorporation of world knowledge, or the handling of lexical entailments and negation. Visitors to the online platform have access to a breakdown of how well each model handles these phenomena alongside its scores on the primary GLUE test sets.

To better understand the challenges posed by the GLUE benchmark, we conduct experiments with simple baselines and state-of-the-art models for sentence representation. We find that naive multi-task learning with standard models over the available task training data yields overall perfor-
Table 1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-Benchmark, which is a regression task. MNLI has three classes while all other classification tasks are binary.

2 Related Work

Our work builds on various strands of NLP research that aspired to develop better general understanding in models.

Multi-task Learning in NLP Multi-task learning has a rich history in NLP as an approach for learning more general language understanding systems. Collobert et al. (2011), one of the earliest works exploring deep learning for NLP, used a multi-task model to jointly learn POS tagging, chunking, named entity recognition, and semantic role labeling. More recently, there has been work using labels from core NLP tasks to supervise training of lower levels of deep neural networks (Søgaard and Goldberg, 2016; Hashimoto et al., 2016) and automatically learning cross-task sharing mechanisms for multi-task learning (Ruder et al., 2017).

Evaluating Sentence Representations Beyond multi-task learning, much of the work so far towards developing general NLU systems has focused on the development of sentence-to-vector encoder functions (Le and Mikolov, 2014; Kiros et al., 2015, i.a.), including approaches leveraging unlabeled data (Hill et al., 2016; Peters et al., 2018), labeled data (Conneau and Kiela, 2018; McCann et al., 2017), and combinations of these (Collobert et al., 2011; Subramanian et al., 2018).
In this line of work, a standard evaluation practice has emerged, and has recently been codified as SentEval (Conneau et al., 2017; Conneau and Kiela, 2018). Like GLUE, SentEval also relies on a variety of existing classification tasks that involve either one or two sentences as inputs, but only evaluates sentence-to-vector encoders. Specifically, SentEval takes a pre-trained sentence encoder as input and feeds its output encodings into lightweight task-specific models (typically linear classifiers) that are trained and tested on task-specific data.

SentEval is well-suited for evaluating general-purpose sentence representations in isolation. However, cross-sentence contextualization and alignment, such as that yielded by methods like soft attention, is instrumental in achieving state-of-the-art performance on tasks such as machine translation (Bahdanau et al., 2014; Vaswani et al., 2017), question answering (Seo et al., 2016; Xiong et al., 2016), and natural language inference. GLUE is designed to facilitate the development of these methods: it is model-agnostic, allowing for any kind of representation or contextualization, including models that use no systematic vector or symbolic representations for sentences whatsoever.

GLUE also diverges from SentEval in the selection of evaluation tasks that are included in the suite. Many of the SentEval tasks are closely related to sentiment analysis, with the inclusion of MR (Pang and Lee, 2005), SST (Socher et al., 2013), CR (Hu and Liu, 2004), and SUBJ (Pang and Lee, 2004). Other tasks are so close to being solved that evaluation on them is less informative, such as MPQA (Wiebe et al., 2005) and TREC (Voorhees et al., 1999). In GLUE, we have attempted to construct a benchmark that is diverse, spans multiple domains, and is systematically difficult.

Evaluation Platforms and Competitions in NLP

Our use of an online evaluation platform with private test labels is inspired by a long tradition of shared tasks at the SemEval (Agirre et al., 2007) and CoNLL (Ellison, 1997) conferences, as well as similar leaderboards on Kaggle and CodaLab. These frameworks tend to focus on a single task, while GLUE emphasizes the need to perform well on multiple different tasks using shared model components.

Weston et al. (2015) similarly proposed a hierarchy of tasks towards building question answering and reasoning models, although involving synthetic language, whereas almost all of our data is human-generated. The recently proposed dialogue systems framework ParlAI (Miller et al., 2017) also combines many language understanding tasks into a single framework, although this aggregation is very flexible, and the framework includes no standardized evaluation suite for system performance.

3 Tasks

We aim for GLUE to spur development of generalizable NLU systems. As such, we expect that doing well on the benchmark should require a model to share substantial knowledge (e.g. in the form of trained parameters) across all tasks, while keeping the task-specific components as minimal as possible. Though it is possible to train a single model for each task and evaluate the resulting set of models on this benchmark, we expect that for some data-scarce tasks in the benchmark, knowledge sharing between tasks will be necessary for competitive performance. In such a case, a more unified approach should prevail.

The GLUE benchmark consists of nine English sentence understanding tasks selected to cover a broad spectrum of task type, domain, amount of data, and difficulty. We describe them here and provide a summary in Table 1. Unless otherwise mentioned, tasks are evaluated on accuracy and have a balanced class split.

The benchmark follows the same basic evaluation model of SemEval and Kaggle. To evaluate a system on the benchmark, one must configure that system to perform all of the tasks, run the system on the provided test data, and upload the results to the website for scoring. The site will then show the user (and the public, if desired) an overall score for the main suite of tasks, and per-task scores on both the main tasks and the diagnostic dataset.

3.1 Single-Sentence Tasks

CoLA The Corpus of Linguistic Acceptability\(^2\) consists of examples of expert English sentence acceptability judgments drawn from 22 books and

---

\(^2\)Available at nyu-mll.github.io/CoLA
journal articles on linguistic theory. Each example is a single string of English words annotated with whether it is a grammatically possible sentence of English. Superficially, this data is similar to our analysis data in that it is constructed to demonstrate potentially subtle and difficult contrasts. However, judgments of this particular kind are the primary form of evidence in linguistic theory (Schütze, 1996), and were a machine learning system to be able to predict them reliably, it would offer potentially substantial evidence on questions of language learnability and innate bias. As in MNLI, the corpus contains development and test examples drawn from in-domain data (the same books and articles used in the training set) and out-of-domain data, though we report numbers only on the unified development and test sets without differentiating these. We follow the original work and report the Matthews correlation coefficient (Matthews, 1975), which evaluates classifiers on unbalanced binary classification tasks with a range from -1 to 1, with 0 being the performance at random chance. We use the standard test set, for which we obtained labels privately from the authors.

STSB The Semantic Textual Similarity Benchmark (Cer et al., 2017) is based on the datasets for a series of annual challenges for the task of determining the similarity on a continuous scale from 1 to 5 of a pair of sentences drawn from various sources. We use the STS-Benchmark release, which draws from news headlines, video and image captions, and natural language inference data, scored by human annotators. We follow common practice and evaluate using Pearson and Spearman correlation coefficients between predicted and ground-truth scores.

3.3 Inference Tasks

MNLI The Multi-Genre Natural Language Inference Corpus (Williams et al., 2018) is a crowdsourced collection of sentence pairs with textual entailment annotations. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis, contradicts the hypothesis, or neither (neutral). The premise sentences are gathered from a diverse set of sources, including transcribed speech, popular fiction, and government reports. The test set is broken into two sections: matched, which is drawn from the same sources as the training set, and mismatched, which uses different sources and thus requires domain transfer. We use the standard test set, for which we obtained labels privately from the authors, and evaluate on both sections.

Though not part of the benchmark, we use and recommend the Stanford Natural Language Inference corpus (Bowman et al. 2015; SNLI) as auxiliary training data. It is distributed in the same format for the same task, and has been used productively in cotraining for MNLI (Chen et al., 2017; Gong et al., 2018).

QNLI The Stanford Question Answering Dataset (Rajpurkar et al. 2016; SQuAD) is a question-answering dataset consisting of question-paragraph pairs, where the one of the sentences in the paragraph (drawn from Wikipedia) contains the answer to the corresponding question (written by an annotator). We automatically convert the original SQuAD dataset into a sentence pair classification task by forming a pair between a question and each sentence in the corresponding context. The task is to determine whether the context sentence contains the answer.
to the question. We filter out pairs where there is low lexical overlap between the question and the context sentence. Specifically, we select all pairs in which the most similar sentence to the question was not the answer sentence, as well as an equal amount of cases in which the correct sentence was the most similar to the question, but another distracting sentence was a close second. This approach to converting pre-existing datasets into NLI format is closely related to recent work by White et al. (2017) as well as to the original motivation for textual entailment presented by Dagan et al. (2006). Both argue that many NLP tasks can be productively reduced to textual entailment. We call this processed dataset QNLI (Question-answering NLI).

RTE The Recognizing Textual Entailment (RTE) datasets come from a series of annual challenges for the task of textual entailment, also known as NLI. We combine the data from RTE1 (Dagan et al., 2006), RTE2 (Bar Haim et al., 2006), RTE3 (Giampiccolo et al., 2007), and RTE5 (Bentivogli et al., 2009). Each example in these datasets consists of a premise sentence and a hypothesis sentence, gathered from various online news sources. The task is to predict if the premise entails the hypothesis. We convert all the data to a two-class split (entailment or not entailment, where we collapse neutral and contradiction into not entailment for challenges with three classes) for consistency.

WNLI The Winograd Schema Challenge (Levesque et al., 2011) is a reading comprehension challenge where each example consists of a sentence containing a pronoun and a list of its possible referents in the sentence. The task is to determine the correct referent. The data is designed to foil simple statistical methods; it is constructed so that each example hinges on contextual information provided by a single word or phrase in the sentence, which can be switched out to change the answer. We use a small evaluation set consisting of new examples derived from fiction books that was shared privately by the authors of the corpus. To convert the problem into a sentence pair classification task, we construct two sentence pairs per example by replacing the ambiguous pronoun with each possible referent. The task (a slight relaxation of the original Winograd Schema Challenge) is to predict if the sentence with the pronoun substituted is entailed by the original sentence. While the included training set is balanced between two classes (entailment and not entailment), the test set is imbalanced between them (35% entailment, 65% not entailment). We call the resulting sentence pair version of the dataset WNLI (Winograd NLI).

3.4 Scoring
In addition to each task’s metric or metrics, Our benchmark reports a macro-average of the metrics over all tasks (see Table 5) to determine a system’s position on the leaderboard. For tasks with multiple metrics (e.g., accuracy and F1), we use unweighted average of the metrics as the score for the task.

3.5 Data and Bias
The tasks listed above are meant to represent a diverse sample of those studied in contemporary research on applied sentence-level language understanding, but we do not endorse the use of the task training sets for any specific non-research application. They do not cover every dialect of English one may wish to handle, nor languages outside of English, and as all of them contain text or annotations that were collected in uncontrolled settings, they contain evidence of stereotypes and biases that one may not wish their system to learn (Rudinger et al., 2017).

4 Diagnostic Dataset
Drawing inspiration from the FraCaS test suite (Cooper et al., 1996) and the recent Build-It-Break-It competition (Ettinger et al., 2017), we include a small, manually-curated test set to allow for fine-grained analysis of system performance on a broad range of linguistic phenomena. While the main benchmarks mostly reflect an application-driven distribution of examples (e.g. the question answering dataset will contain questions that people are likely to ask), our diagnostic dataset is collected to highlight a pre-defined set of modeling-relevant phenomena.

Specifically, we construct a set of NLI examples with fine-grained annotations of the linguistic phe-
<table>
<thead>
<tr>
<th>LS</th>
<th>PAS</th>
<th>L</th>
<th>K</th>
<th>Sentence 1</th>
<th>Sentence 2</th>
<th>Fwd</th>
<th>Bwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>Cape sparrows eat seeds, along with soft plant parts and insects.</td>
<td>Seeds, along with soft plant parts and insects, are eaten by cape sparrows.</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>Cape sparrows eat seeds, along with soft plant parts and insects.</td>
<td>Cape sparrows are eaten by seeds, along with soft plant parts and insects.</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Tulsi Gabbard disagrees with Bernie Sanders on what is the best way to deal with Bashar al-Assad.</td>
<td>Tulsi Gabbard and Bernie Sanders disagree on what is the best way to deal with Bashar al-Assad.</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Musk decided to offer up his personal Tesla roadster.</td>
<td>Musk decided to offer up his personal car.</td>
<td>E</td>
<td>N</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>The announcement of Tillerson’s departure sent shock waves across the globe.</td>
<td>People across the globe were not expecting Tillerson’s departure.</td>
<td>E</td>
<td>N</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>The announcement of Tillerson’s departure sent shock waves across the globe.</td>
<td>People across the globe were prepared for Tillerson’s departure.</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>I have never seen a hummingbird not flying.</td>
<td>I have never seen a hummingbird.</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>Understanding a long document requires tracking how entities are introduced and evolve over time.</td>
<td>Understanding a long document requires evolving over time.</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>Understanding a long document requires tracking how entities are introduced and evolve over time.</td>
<td>Understanding a long document requires understanding how entities are introduced.</td>
<td>E</td>
<td>N</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>That perspective makes it look gigantic.</td>
<td>That perspective makes it look minuscule.</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

Table 2: Examples from the analysis set. Sentence pairs are labeled according to four coarse categories: Lexical Semantics (L), Predicate-Argument Structure (PAS), Logic (L), and Knowledge and Common Sense (K). Within each category, each example is also tagged with fine-grained labels (see tables 4). See gluebenchmark.com for details on the set of labels, their meaning, and how we do the categorization.

nomina they capture. The NLI task is well suited to this kind of analysis, as it is constructed to make it straightforward to evaluate the full set of skills involved in (ungrounded) sentence understanding, from the resolution of syntactic ambiguity to pragmatic reasoning with world knowledge. We ensure that the examples in the diagnostic dataset have a reasonable distribution over word types and topics by building on naturally-occurring sentences from several domains. Table 2 shows examples from the dataset.

Linguistic Phenomena We tag every example with fine- and coarse-grained categories of the linguistic phenomena they involve (categories shown in Table 3). While each example was collected with a single phenomenon in mind, it is often the case that it falls under other categories as well. We therefore code the examples under a non-exclusive tagging scheme, in which a single example can participate in many categories at once. For example, to know that I like some dogs entails I like some animals, it is not sufficient to know that dog lexically entails animal; one must also know that dog/animal appears in an upward monotone context in the sentence. This example would be classified under both Lexical Semantics > Lexical Entailment and Logic > Upward Monotone.

Domains We construct sentences based on existing text from four domains: News (drawn from articles linked on Google News), Reddit (from threads linked on the Front Page), Wikipedia (from Featured Articles), and academic papers drawn from the proceedings of recent ACL conferences. We include 100 sentence pairs constructed from each source, as well as 150 artificially-constructed sentence pairs.

Annotation Process We begin with an initial set of fine-grained semantic phenomena, using the

---

7 news.google.com
8 reddit.com
9 en.wikipedia.org/wiki/Wikipedia:Featured_articles
Table 3: The types of linguistic phenomena annotated in the diagnostic dataset, organized under four major categories.

<table>
<thead>
<tr>
<th>Coarse-Grained Categories</th>
<th>Fine-Grained Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical Semantics</td>
<td>Lexical Entailment, Morphological Negation, Factivity, Symmetry/Collectivity, Redundancy, Named Entities, Quantifiers</td>
</tr>
<tr>
<td>Predicate-Argument Structure</td>
<td>Core Arguments, Prepositional Phrases, Ellipsis/Implicits, Anaphora/Coreference Active/Passive, Nominalization, Genitives/Partitives, Datives, Relative Clauses, Coordination Scope, Intersectivity, Restrictivity</td>
</tr>
<tr>
<td>Logic</td>
<td>Negation, Double Negation, Intervals/Numbers, Conjunction, Disjunction, Conditionals, Universal, Existential, Temporal, Upward Monotone, Downward Monotone, Non-Monotone</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Common Sense, World Knowledge</td>
</tr>
</tbody>
</table>

categories in the FraCaS test suite (Cooper et al., 1996) as a starting point, while also generalizing to include lexical semantics, common sense, and world knowledge. We gather examples by searching through text in each domain and locating example sentences that can be easily modified to involve one of the chosen phenomena (or that involves one already). We then modify the sentence further to produce the other sentence in an NLI pair. In many cases, we make these modifications small, in order to encourage high lexical and structural overlap among the sentence pairs—which may make the examples more difficult for models that rely on lexical overlap as an indicator for entailment. We then label the NLI relations between the sentences in both directions (considering each sentence alternatively as the premise), producing two labeled examples for each pair. Where possible, we produce several pairs with different labels for a single sentence, to have minimal sets of sentence pairs that are lexically and structurally very similar but correspond to different entailment relationships. After finalizing the categories, we gathered a minimum number of examples in each fine-grained category from each domain to ensure a baseline level of diversity.

In total, we gather 550 sentence pairs, for 1100 entailment examples. The labels are 42% entailment, 35% neutral, and 23% contradiction.

Auditing In light of recent work showing that crowdsourced data often contains artifacts which can be exploited to perform well without solving the intended task (Schwartz et al., 2017; Gururangan et al., 2018), we perform an audit of our manually curated data as a sanity check. We reproduce the methodology of Gururangan et al. (2018), training fastText classifiers (Joulin et al., 2016) to predict entailment labels on SNLI and MultiNLI using only the hypothesis as input. Testing these on the diagnostic data, accuracies are 32.7% and 36.4%—very close to chance—showing that the data does not suffer from artifacts of this specific kind. We also evaluate state-of-the-art NLI models on the diagnostic dataset and find their overall performance to be rather weak, further suggesting that no easily-gameable artifacts present in existing training data are abundant in the diagnostic dataset (see Section 6).

Evaluation Since the class distribution in the diagnostic set is not uniform (and is even less so within each category), we propose using $R_3$, a three-class generalization of the Matthews correlation coefficient, as the evaluation metric. This coefficient was introduced by Gorodkin (2004) as $R_K$, a generalization of the Pearson correlation that works for $K$ dimensions by averaging the square error from the mean value in each dimension, i.e., calculating the full covariance between the input and output. In the discrete case, it generalizes Matthews correlation, where a value of 1 means perfect correlation and 0 means random chance.

Intended Use Because these analysis examples are hand-picked to address certain phenomena, we expect that they will not be representative of the distribution of language as a whole, even in the targeted domains. However, NLI is a task with no natural input distribution. We deliberately select sentences that we hope will be able to provide insight into what models are doing, what phenomena they catch on to, and where are they limited. This means that the raw performance numbers on the analysis set should be taken with a grain of salt. The set is provided not as a benchmark, but as an analysis tool to paint in broad strokes the kinds
or phenomena a model may or may not capture, and to provide a set of examples that can serve for error analysis, qualitative model comparison, and development of adversarial examples that expose a model’s weaknesses.

5 Baselines

As baselines, we provide performance numbers for a relatively simple multi-task learning model trained from scratch on the benchmark tasks, as well as several more sophisticated variants that utilize recent developments in transfer learning. We also evaluate a sample of competitive existing sentence representation models, where we only train task-specific classifiers on top of the representations they produce.

5.1 Multi-task Architecture

Our simplest baseline is based on sentence-to-vector encoders, and sets aside GLUE’s ability to evaluate models with more complex structures. Taking inspiration from Conneau et al. (2017), the model uses a BiLSTM with temporal max-pooling and 300-dimensional GloVe word embeddings (Pennington et al., 2014) trained on 840B Common Crawl. For single-sentence tasks, we process the sentence and pass the resulting vector to a classifier. For sentence-pair tasks, we process sentences independently to produce vectors \( u \), \( v \), and pass \( [u; v; u - v; u + v] \) to a classifier. We experiment with logistic regression and a multi-layer perceptron with a single hidden layer for classifiers, leaving the choice as a hyperparameter to tune.

For sentence-pair tasks, we take advantage of GLUE’s indifference to model architecture by incorporating a matrix attention mechanism between the two sentences. By explicitly modeling the interaction between sentences, our model is strictly outside of the sentence-to-vector paradigm. We follow standard practice to contextualize each token with attention. Given two sequences of hidden states \( u_1, u_2, \ldots, u_M \) and \( v_1, v_2, \ldots, v_N \), the attention mechanism is implemented by first computing a matrix \( H \) where \( \forall i \in \{1, 2, \ldots, M\}, \forall j \in \{1, 2, \ldots, N\}, H_{ij} = u_i \cdot v_j \). For each \( u_i \), we get attention weights \( \alpha_{ij} \) by taking a softmax over the \( i^{th} \) row of \( H \), and get the corresponding context vector \( \tilde{v}_i = \sum_j \alpha_{ij} v_j \) by taking the attention-weighted sum of the \( v_j \). We pass a second BiLSTM with max pooling over the sequence \( [u_1; v_1; \ldots; u_M; v_M] \) to produce \( u' \). We process the \( v_j \) vectors in a symmetric manner to obtain \( v' \). Finally, we feed \( [u'; v'; u' - v'; u' + v'] \) into a classifier for each task.

Incorporating Transfer Learning We also augment our base non-attentive model with two recently proposed methods for transfer learning in NLP: ELMo (Peters et al., 2018) and CoVe (McCann et al., 2017). Both use pretrained models that produce contextual word embeddings via some transformation of the underlying model’s hidden states.

ELMo uses a pair of two-layer neural language models (one forward, one backward) trained on the One Billion Word Benchmark (Chelba et al., 2013). A word’s contextual embedding is produced by taking a linear combination of the corresponding hidden states on each layer. We follow the authors’ recommendations\(^{10}\) and use the ELMo embeddings in place of any other embed-
Table 5: Performances on the benchmark tasks for different models. Bold denotes best results per task overall; underline denotes best results per task within a section; A denotes models using attention. For MNLI, we report accuracy on the matched / mismatched test splits. For MRPC and Quora, we report accuracy / F1. For STS-B, we report Pearson / Spearman correlation, scaled to be in [-100, 100]. For CoLA, we report Matthews correlation, scaled to be in [-100, 100]. For all other tasks we report accuracy (%). We compute a macro-average score in the style of SentEval by taking the average across all tasks, first averaging the metrics within each tasks for tasks with more than one reported metric.

5.2 Multi-task Training

These four models (BiLSTM, BiLSTM +Attn, BiLSTM +ELMo, BiLSTM +CoVe) are jointly trained on all tasks, with the primary BiLSTM encoder shared between all task-specific classifiers. To perform multi-task training, we randomly pick an ordering on the tasks and train on 10% of a task’s training data for each task in that order. We repeat this process 10 times between validation checks, so that we roughly train on all training examples for each task once between checks. We use the previously defined macro-average as the validation metric, where for tasks without predetermined development sets, we reserve 10% of the training data for validation.

We train our models with stochastic gradient descent using batch size 128, and multiply the learning rate by .2 whenever validation performance does not improve. We stop training when the learning rate drops below $10^{-5}$ or validation performance does not improve after 5 evaluations.

We tune hyperparameters with random search over 30 runs on macro-average development set performance. Our best model is a two layer BiLSTM that is 1500-dimensional per direction. We evaluate our all our BiLSTM-based models with these settings.

5.3 Single-task Training

We use the same training procedure to train an instance of the model with ELMo on each task separately. For tuning hyperparameters per task, we use random search on that task’s metrics evaluated on the development set. We tune the same hyperparameters as in the multi-task setting, except we also tune whether or not to use attention (for pair tasks only), and whether to use SGD or Adam (Kingma and Ba, 2014).

5.4 Sentence Representation Models

Finally, we evaluate a number of established sentence-to-vector encoder models using our suite. Specifically, we investigate:

1. CBoW: the average of the GloVe embeddings of the tokens in the sentence.

2. Skip-Thought (Kiros et al., 2015): a sequence-to-sequence(s) model trained to generate the previous and next sentences given the middle sentence. After training, the
model’s encoder is taken as a sentence encoder. We use the original pretrained model\(^\text{11}\) trained on sequences of sentences from the Toronto Book Corpus (Zhu et al. 2015, TBC).

3. InferSent (Conneau et al., 2017): a BiLSTM with max-pooling trained on MNLI and SNLI.

4. DisSent (Nie et al., 2017): a BiLSTM with max-pooling trained to predict the discourse marker (e.g. “because”, “so”, etc.) relating two sentences on data derived from TBC (Zhu et al., 2015). We use the variant trained to predict eight discourse marker types.

5. GenSen (Subramanian et al., 2018): a sequence-to-sequence model trained on a variety of supervised and unsupervised objectives. We use a variant of the model trained on both MNLI and SNLI, the Skip-Thought objective on TBC, and a constituency parsing objective on the One Billion Word Benchmark.

We use pretrained versions of these models, fix their parameters, learn task-specific classifiers on top of the sentence representations that they produce. We use the SentEval framework to train the classifiers.

### 6 Benchmark Results

We present performance on the main benchmark in Table 5. For multi-task models, we average performance over five runs; for single-task models, we use only one run.

We find that the single-task baselines have the best performance among all models on SST-2, MNLI, and QNLI, while the lagging behind multi-task trained models on MRPC, STS-B, and RTE. For MRPC and RTE in particular, the single-task baselines are close to majority class baselines, indicating the inherent difficulty of these tasks and the potential of transfer learning approaches. On QQP, the best multi-task trained models slightly outperform the single-task baseline.

For multi-task trained baselines, we find that almost no model does significantly better on CoLA or WNLI than performance from predicting majority class (0.0 and 63, respectively), which highlights the difficulty of current models to generalize to these tasks. The notable exception is DisSent, which does better than other multi-task models on CoLA. A possible explanation is that DisSent is trained using a discourse-based objective, which might be more sensitive to grammaticality. However, DisSent underperforms other multi-task models on more data-rich tasks such as MNLI and QNLI. This result demonstrates the utility of GLUE: by assembling a wide variety of tasks, it highlights the relative strengths and weaknesses of various models.

Among our multi-task BiLSTM models, using attention yields a noticeable improvement over the vanilla BiLSTM for all tasks involving sentence pairs. When using ELMo or CoVe, we see improvements for nearly all tasks. There is also a performance gap between all variants of our multi-task BiLSTM model and the best models that use pre-trained sentence representations (GenSen and InferSent), demonstrating the utility of transfer via pre-training on an auxiliary task.

Among the pretrained sentence representation models, we observe relatively consistent per-task and aggregate performance gains moving from CBoW to Skip-Thought to DisSent, to InferSent and GenSen. The latter two show competitive performance on various tasks, with GenSen slightly edging out InferSent in aggregate.

### 7 Analysis

By running all of the models on the diagnostic set, we get a breakdown of their performance across a set of modeling-relevant phenomena. Overall re-
<table>
<thead>
<tr>
<th>Gold \ Prediction</th>
<th>All</th>
<th>E</th>
<th>C</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>65</td>
<td>16</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>42</td>
<td>34</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>23</td>
<td>11</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>N</td>
<td>35</td>
<td>19</td>
<td>5</td>
<td>11</td>
</tr>
</tbody>
</table>

(a) Confusion matrix for BiLSTM +Attn (percentages).

<table>
<thead>
<tr>
<th>Model</th>
<th>UQuant</th>
<th>MNeg</th>
<th>2Neg</th>
<th>Coref</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>67</td>
<td>13</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td>BiLSTM +Attn</td>
<td>85</td>
<td>64</td>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>BiLSTM +ELMo</td>
<td>77</td>
<td>60</td>
<td>-8</td>
<td>18</td>
</tr>
<tr>
<td>BiLSTM +CoVe</td>
<td>71</td>
<td>34</td>
<td>28</td>
<td>39</td>
</tr>
<tr>
<td>CBoW</td>
<td>16</td>
<td>0</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>SkipThought</td>
<td>61</td>
<td>6</td>
<td>-2</td>
<td>30</td>
</tr>
<tr>
<td>InferSent</td>
<td>64</td>
<td>51</td>
<td>-22</td>
<td>26</td>
</tr>
<tr>
<td>DisSent</td>
<td>70</td>
<td>34</td>
<td>-20</td>
<td>21</td>
</tr>
<tr>
<td>GenSen</td>
<td>78</td>
<td>64</td>
<td>5</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 7: Model performance in terms of $R_3$ (scaled by 100) on selected fine-grained categories for analysis. The categories are Universal Quantification (UQuant), Morphological Negation (MNeg), Double Negation (2Neg), and Anaphora/Coreference (Coref).

(b) Output class distributions (percentages). Bolded numbers are closest to the gold distribution.

Figure 1: Partial output of GLUE’s error analysis, aggregated across our models.

Results are presented in Table 6.

Overall Performance Performance is very low across the board: the highest total score (27) still denotes poor absolute performance. Scores on the Predicate-Argument Structure category tend to be higher across all models, while Knowledge category scores are lower. However, these trends do not necessarily reflect that our models understand sentence structure better than world knowledge or common sense; these numbers are not directly comparable. Rather, numbers should be compared between models within each category.

One notable trend is the high performance of the BiLSTM +Attn model: though it does not outperform most of the pretrained sentence representation methods (InferSent, DisSent, GenSen) on GLUE’s main benchmark tasks, it performs best or competitively on all categories of the diagnostic set.

Domain Shift & Class Priors GLUE’s online platform also provides a submitted model’s predicted class distributions and confusion matrices. We provide an example in Figure 1. One point is immediately clear: all models severely underpredict neutral and over-predict entailment. This is perhaps indicative of the models’ inability to generalize and adapt to new domains. We hypothesize that they learned to treat high lexical overlap as a strong sign of entailment, and that surgical addition of new information to the hypothesis (as in the case of neutral instances in the diagnostic set) might go unnoticed. Indeed, the attention-based model seems more sensitive to the neutral class, and is perhaps better at detecting small sets of unaligned tokens because it explicitly tries to model these alignments.

Linguistic Phenomena While performance metrics on the coarse-grained categories give us broad strokes that we can use to compare models, we can gain a better understanding of the models’ capabilities by drilling down into the fine-grained subcategories. The GLUE platform reports scores for every fine-grained category; we present here a few highlights in Table 7. To help interpret these results, we list some examples from each fine-grained category, along with model predictions, in Table 4.

The Universal Quantification category appears easy for most of the models; looking at examples, it seems that when universal quantification as a phenomenon is isolated, catching on to lexical cues such as all often suffices to solve our examples. Morphological negation examples are superficially similar, but the systems find it more difficult. On the other hand, double negation appears to be adversarially difficult for models to recognize, with the exception of BiLSTM +CoVe; this is perhaps due to the translation signal, which can match phrases like “not bad” and “okay” to the same expression in a foreign language. A similar advantage, though less acute, appears when using CoVe on coreference examples.
Overall, there is some evidence that going beyond sentence-to-vector representations might aid performance on out-of-domain data (as with BiLSTM +Attn) and that representations like ELMo and CoVe encode important linguistic information that is specific to their supervision signal. Our platform and diagnostic dataset should support future inquiries into these issues, so we can better understand our models’ generalization behavior and what kind of information they encode.

8 Conclusion

We introduce GLUE, a platform and collection of resources for training, evaluating, and analyzing general natural language understanding systems. When evaluating existing models on the main GLUE benchmark, we find that none are able to substantially outperform a relatively simple baseline of training a separate model for each constituent task. When evaluating these models on our diagnostic dataset, we find that they spectacularly fail on a wide range of linguistic phenomena. The question of how to design general-purpose NLU models thus remains unanswered. We believe that GLUE, and the generality it promotes, can provide fertile soil for addressing this open challenge.

Acknowledgments

We thank Ellie Pavlick, Tal Linzen, Kyunghyun Cho, and Nikita Nangia for their comments on this work at its early stages, and we thank Ernie Davis, Alex Warstadt, and Quora’s Nikhil Danekar and Kornel Csernai for providing access to private evaluation data. This project has benefited from financial support to SB by Google, Tencent Holdings, and Samsung Research.

References


