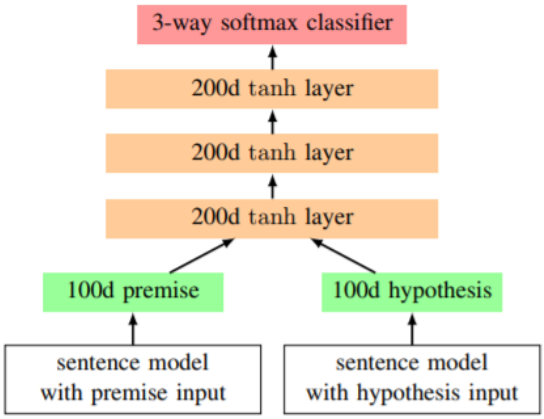
Corpus: https://nlp.stanford.edu/projects/snli/
Excitement Open Platform: https://github.com/hltfbk/EOP-1.2.3/wiki/Installation

- NLI has been addressed using a variety of techniques, including those based on symbolic logic, **knowledge bases**, and **neural networks**.
- **Distributed representations** excel at **capturing relations based in similarity**, and have proven effective at modeling simple dimensions of meaning like evaluative sentiment (e.g., Socher et al. 2013), but it is less clear that they can be trained to support the full range of logical and commonsense inferences required for NLI (Bowman et al., 2015; Weston et al., 2015b; Weston et al., 2015a).
- Datasets available are generally **too small** for training modern data-intensive, wide-coverage models, many **contain sentences that were algorithmically generated**, and they are often **beset with indeterminacies of event and entity coreference** that significantly impact annotation quality
- In this paper, we use this corpus to **evaluate a variety of models** for natural language inference, including **rule-based systems, simple linear classifiers, and neural network-based models.**
- We find that two models achieve comparable performance: **a feature-rich classifier** model and a neural network model centered around a **Long Short-Term Memory network**
- We further evaluate the LSTM model by taking advantage of its **ready support for transfer learning**, and show that it can be adapted to an existing NLI challenge task, yielding the best reported performance by a neural network model and approaching the overall state of the art.
- Qualitatively, we find the data that we collected draws fairly extensively on commonsense knowledge, and that **hypothesis and premise sentences often differ structurally in significant ways**, suggesting that there is **room for improvement** beyond superficial word alignment models
- The test and development sets contain **10k examples** each.
- The first class of models is from the **Excitement Open Platform** (EOP, Pado et al. 2014 ´ ; Magnini et al. 2014)—an open source platform for RTE research. We evaluate on **two algorithms included in the distribution**: a **simple edit-distance based algorithm** and a **classifier-based algorithm**, the latter both **in a bare form** and **augmented with EOP's full suite of lexical resources.**
- We approached this by **running the same system on several data sets**: our own test set, the SICK test data, and the standard RTE-3 test set
- All models are evaluated only on **2-class entailment**. To convert 3-class problems like SICK and SNLI to this setting, all instances of **contradiction and unknown are converted to nonentailment**

- This yields a most-frequent-class **baseline accuracy of 66% on SNLI, and 71% on SICK**
- The edit distance algorithm **tunes the weight of the three case insensitive edit distance operations** on the training set, **after removing stop words**. In addition to the base classifier-based system distributed with the platform, we train **a variant which includes information from WordNet** (Miller, 1995) **and VerbOcean** (Chklovski and Pantel, 2004), and **makes use of features based on tree patterns and dependency tree skeletons** (Wang and Neumann, 2007)
- Our **classifier** implements **6 feature types; 3 unlexicalized and 3 lexicalized**:
    - The BLEU score of the hypothesis with respect to the premise, using an n-gram length between 1 and 4.
    - The length difference between the hypothesis and the premise, as a real-valued feature.
    - The overlap between words in the premise and hypothesis, both as an absolute count and a percentage of possible overlap, and both over all words and over just nouns, verbs, adjectives, and adverbs.
    - An indicator for every unigram and bigram in the hypothesis.
    - Cross-unigrams: for every pair of words across the premise and hypothesis which share a POS tag, an indicator feature over the two words.
    - for every pair of bigrams across the premise and hypothesis which share a POS tag on the second word, an indicator feature over the two bigrams.
- On our large corpus in particular, there is a **substantial jump in accuracy from using lexicalized features,** and **another from using the very sparse cross-bigram features**. The latter result suggests that **there is value in letting the classifier automatically learn** to recognize structures like explicit negations and adjective modification.
- Although we expect that richer models would perform better, the results suggest that **given enough data, cross bigrams with the noisy part-of-speech overlap constraint can produce an effective model**.
- Each neural network must produce a vector representation of each of the two sentences without using any context from the other sentence, and the two resulting vectors are then passed to a neural network classifier which predicts the label for the pair.
- The neural network classification architecture: for each sentence embedding model evaluated in Tables 6 and 7, two identical copies of the model are run with the two

sentences as input, and their outputs are used as the two 100d inputs shown here.



- Our **neural network classifier**, depicted in Figure 3 (and based on a one-layer model in Bowman et al. 2015), is simply **a stack of three 200d tanh layers**, with the **bottom layer taking the concatenated sentence** representations as input and the **top layer feeding a softmax classifier**, all **trained jointly with the sentence embedding model** itself.
- We test **three sentence embedding models**, each set to use **100d phrase and sentence embeddings.** Our **baseline sentence embedding model** simply **sums the embeddings of the words in each sentence**. In addition, we experiment with **two simple sequence embedding models**: a plain **RNN** and an **LSTM RNN** (Hochreiter and Schmidhuber, 1997).
- The **word embeddings** for all of the models are **initialized with the 300d reference** GloVe vectors (840B token version, Pennington et al. 2014) and fine-tuned as part of training
- In addition, all of the models use **an additional tanh neural network layer** to map these 300d embeddings into the lower-dimensional phrase and sentence embedding space. All of the models are **randomly initialized using standard techniques** and **trained using AdaDelta** (Zeiler, 2012) **minibatch SGD** until performance on the development set stops improving.
- We applied **L2 regularization to all models**, manually tuning the strength coefficient λ for each, and **additionally applied dropout** (Srivastava et al., 2014) to the **inputs and outputs of the sentence embedding models** (though not to its internal connections) with a fixed dropout rate.
- The **sum of words model performed slightly worse** than the fundamentally similar lexicalized classifier— while the sum of words model can use pretrained word embeddings to better handle rare words, it **lacks even the rudimentary sensitivity to word order** that the lexicalized model's bigram features provide
- While the lexicalized model fits the training set almost perfectly, **the gap between train and test set accuracy is relatively small for all three neural network** models,

suggesting that research into significantly higher capacity versions of these models would be productive.

- Despite the large size of the training corpus and the distributional information captured by GloVe initialization, **many lexical relationships are still misanalyzed**, leading to incorrect predictions of *independent*, even for pairs that are common in the training corpus like *beach/surf* and *sprinter/runner.*
- Semantic mistakes at the phrasal level (e.g., predicting contradiction for A male is placing an order in a deli/A man buying a sandwich at a deli) indicate that **additional attention to compositional semantics would pay off**.
- Another headline example of this type is A man wearing padded arm protection is being bitten by a German shepherd dog/A man bit a dog, which all the models wrongly diagnose as entailment, though the sentences report two very different stories. **A model with access to explicit information about syntactic or semantic structure should perform better on cases like these.**
- To perform transfer, we **take the parameters of the LSTM RNN model trained on SNLI** and use them to **initialize a new model**, which is **trained from that point only on the training portion of SICK.**
- The only newly **initialized parameters are softmax** layer parameters and the **embeddings** for words that appear in SICK, but not in SNLI (which are populated with GloVe embeddings as above).
- We use the **same model hyperparameters** that were used to train the original model, **with the exception of the L2 regularization** strength, which is re-tuned.
- In contrast, transferring SNLI representations to SICK yields the best performance yet reported for an unaugmented neural network model, surpasses the available EOP models, and approaches both the overall state of the art at 84.6% (Lai and Hockenmaier, 2014) and the 84% level of interannotator agreement, which likely represents an approximate performance ceiling.
- Further research on effective transfer learning on small data sets with neural models might facilitate improvements here.
- This paper sought to remedy this with a new, largescale, naturalistic corpus of sentence pairs labeled for entailment, contradiction, and independence.
- We used this corpus to evaluate a range of models, and found that both simple lexicalized models and neural network models perform well, and that the representations learned by a neural network model on our corpus can be used to dramatically improve performance on a standard challenge dataset.