# WTMED at MEDIQA 2019: A Hybrid Approach to Biomedical Natural Language Inference

**Zhaofeng Wu**
Paul G. Allen School of CSE
University of Washington
zfw7@cs.washington.edu

**Yan Song**
Tencent AI Lab
clksong@gmail.com

**Sicong Huang**
Department of ECE
University of Washington
huangs33@uw.edu

**Yuanhe Tian**
Department of Linguistics
University of Washington
yhtian@uw.edu

**Fei Xia**
Department of Linguistics
University of Washington
fxia@uw.edu

## Abstract

Natural language inference (NLI) is challenging, especially when it is applied to technical domains such as biomedical settings. In this paper, we propose a hybrid approach to biomedical NLI where different types of information are exploited for this task. Our base model includes a pre-trained text encoder as the core component, and a syntax encoder and a feature encoder to capture syntactic and domain-specific information. Then we combine the output of different base models to form more powerful ensemble models. Finally, we design two conflict resolution strategies when the test data contain multiple (premise, hypothesis) pairs with the same premise. We train our models on the MedNLI dataset, yielding the best performance on the test set of the MEDIQA 2019 Task 1.

## 1 Introduction

Natural language inference (NLI) (MacCartney and Manning, 2009), also known as textual entailment, is an important natural language processing (NLP) task that has long been studied (Bowman et al., 2015; Parikh et al., 2016; Chen et al., 2016; Conneau et al., 2017; Tay et al., 2018). It aims to capture the relationship between two sentences, identifying whether a given *premise* entails, contradicts, or is neutral to a given *hypothesis*. Success in NLI is crucial for achieving semantic comprehension of human language, which in turn is a prerequisite to accomplish natural language understanding (NLU). In general, accurate NLI systems facilitate many downstream tasks, such as commonsense reasoning (Zellers et al., 2018) and question answering (Abacha and Demner-Fushman, 2016, 2017).

Most of existing NLI studies are conducted in the general domain (Marelli et al., 2014; Bowman et al., 2015; Williams et al., 2018), with limited attention paid to domain-specific scenarios. Nevertheless, there has been increasing demand for information processing in the biomedical domain such as biomedical question answering (Abacha and Demner-Fushman, 2019) and cohort selection (Glicksberg et al., 2018). Many biomedical NLP applications require automatic understanding of symptom descriptions and examination reports (Abacha and Demner-Fushman, 2016, 2017) and therefore can greatly benefit from accurate biomedical NLI systems.

In this study, we propose a hybrid approach to biomedical NLI, which includes three main components, as illustrated in Figure 1. The main component is the base model (the largest box in the figure), which includes three encoders: an MT-DNN (Liu et al., 2019c) based text encoder, a syntax encoder that captures structural information, and a feature encoder which injects some degree of domain knowledge into the model (see §3). We conduct unsupervised pre-training for the text encoder on biomedical corpora to compensate for the lack of domain-specific supervision (Lee et al., 2019). To enhance our model, we also use model ensemble and conflict resolution strategies, corresponding to the two top dashed boxes in Figure 1 and are explained in §4. The datasets and implementation detail are described in §5. The experimental results on the MedNLI dataset (Romanov and Shivade, 2018) and the MEDIQA 2019 shared task 1 (Ben Abacha et al., 2019) are reported in §6.[1]

## 2 Related Work

A common neural network approach to address the NLI task is sentence pair modeling (Lan and

[1] Our code is publicly available at https://github.com/ZhaofengWu/MEDIQA_WTMED

*Proceedings of the BioNLP 2019 workshop*, pages 415–426
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

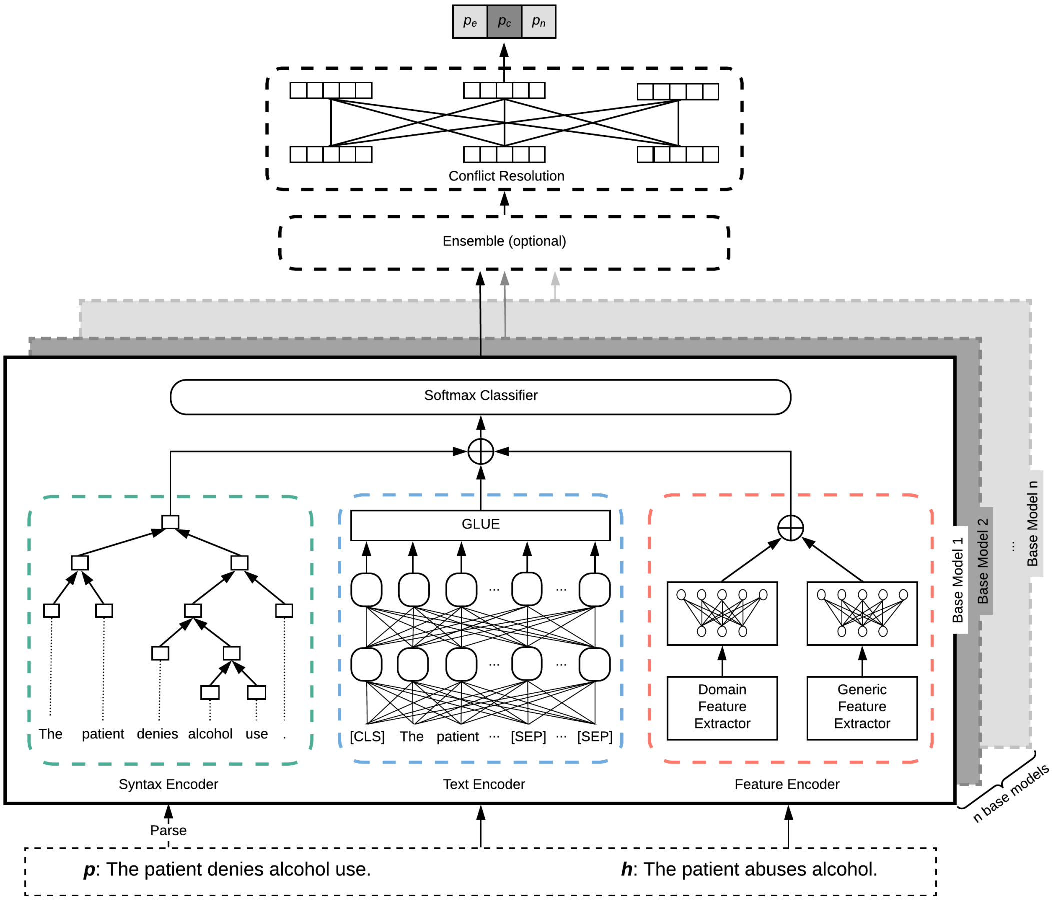

Figure 1: Our overall system. Our base model consists of three encoders and a softmax classifier: a syntax encoder that encodes the constituency parses provided by the dataset to a vector representation via Tree-LSTM; an MT-DNN based text encoder; a feature encoder that encodes domain and generic string-based features through fully-connected layers; and a softmax classifier that takes in the concatenation (⊕) of the three encoders' output and generates a prediction. The output of base models is sent to the ensemble and conflict resolution modules (the multimodal attention method is depicted here as an example) to make a final prediction.

Xu, 2018). The premise and hypothesis are separately embedded (e.g. via GloVe (Pennington et al., 2014) or ELMo (Peters et al., 2018)) and encoded (e.g. via CNN or LSTM). Typically an interaction layer is employed to add information alignment between the premise and the hypothesis. For example, between the two baseline models used in the MedNLI dataset, InferSent (Conneau et al., 2017) computes the interaction vector via $[p; h; |p - h|; p * h]$ and ESIM (Chen et al., 2016) uses an attention matrix to softly align the two representations. ESIM also appends an inference composition layer to propagate the local attended information. A softmax layer is used to classify the final representation.

The recent Transformer-based models have been demonstrated to be a better encoder at NLI than CNN and LSTM by fully attending over the two sentences (Radford, 2018; Devlin et al., 2018). BERT (Devlin et al., 2018) pre-trains the model with large unlabeled corpora which allows better text representations. MT-DNN (Liu et al., 2019c) leverages multi-task learning (Liu et al., 2015) to fine-tune the BERT weights using the GLUE datasets (Wang et al., 2018). The authors showed that resulting representations outperform BERT on many NLU tasks.

On top of this sentence pair modeling scheme, previous studies have independently leveraged syntax (Chen et al., 2016), external knowledge (Chen et al., 2018; Lu et al., 2019), ensemble methods (Ghaeini et al., 2018b), and language model fine-tuning (Alsentzer et al., 2019) to improve the performance of NLI systems. Nonetheless, to our knowledge, there have been no empirical results on the effect of combining these additions simultaneously. Additionally, as recent studies have pointed out that pre-trained contextualized representations contain rich linguistic signals (Hewitt and Manning, 2019; Liu et al., 2019b), it is reasonable to ask whether explicitly integrating knowledge will continue to augment such representations. Our work can be seen as an empirical study to examine the efficacy of applying multiple additions on top of Transformer-based models.

## 3 Base Model

NLI is generally treated as a three-way classification task that models whether a given premise $p$ entails, contradicts, or is neutral to a hypothesis $h$. A classifier $f$ is learned taking $p$ and $h$ as input to predict the class probabilities

$$f(p, h) = \begin{bmatrix} P_e & P_c & P_n \end{bmatrix}^\top \tag{1}$$

with $P_r; r \in \{e, c, n\}$ representing the probability for *entailment*, *contradiction*, and *neutral*. The final result is the class with the highest probability

$$y_{p,h} = \underset{r \in \{e,c,n\}}{\arg\max}\, P_r \tag{2}$$

As illustrated in Figure 1, our base model contains three modules. The widely used pre-trained Transformer model (Devlin et al., 2018; Liu et al., 2019c) serves as the basic text encoder to represent $p$ and $h$. A syntax encoder and a feature encoder are also utilized to augment the basic representation by extracting and encoding more information from the input. The details of these encoders and how they are combined for $f$ are discussed in the following subsections.

### 3.1 Text Encoder

Text representation is crucial to facilitate downstream tasks (Song et al., 2017, 2018). As a part of recent advancements in NLP, pre-trained models provide strong baselines for sentence representations and allow great generalizability for the represented text. Therefore, to represent $p$ and $h$, we adopt a pre-trained Transformer model, MT-DNN (Liu et al., 2019c), as the text encoder in our base model. MT-DNN is based on BERT (Devlin et al., 2018) and additionally fine-tuned on GLUE (Wang et al., 2018), a set of NLU datasets including NLI subsets. Through its multi-task learning objective, MT-DNN allows a more general and powerful representation for natural language understanding than BERT (Liu et al., 2019c). Formally, one can briefly describe the encoder as

$$\mathcal{V}_{TE}(p, h) = \text{MT-DNN}(p, h) \tag{3}$$

with $\mathcal{V}_{TE}(p, h)$ referring to the output of the text encoder, a vector representing $p$ and $h$.

Pre-training on large unlabeled corpora with a language modeling objective has facilitated many recent state-of-the-art advancements (Peters et al., 2018; Radford, 2018; Devlin et al., 2018; Lee et al., 2019; Radford et al., 2019). Inspired by these results, we enhance the MT-DNN representation by further fine-tuning on unlabeled biomedical data to mitigate the lack of in-domain supervision.

### 3.2 Syntax Encoder

Linguistic understandings, for example coreference relations (Zhang et al., 2019a,b), could aid the interpretation of a sentence. Syntactic structures are often useful for deciding the entailment of a sentence pair (Chen et al., 2016). There exist numerous NLI examples where a hypothesis is merely the premise with adjunct phrases removed. The syntax encoder also mitigates the out-of-vocabulary issue which is common in specific domains (Liu et al., 2019a) by capturing the structural information. Therefore, we include a syntax encoder in our base model.

We use Tree-LSTM (Tai et al., 2015) to model constituency parse trees of $p$ and $h$. For each sentence, we encode it according to its tree structure and take the final state of the root node to represent the entire sentence. Formally, taking $p$ as an example, the syntax encoder can be formulated as

$$\mathcal{V}_{SE}(p) = \text{Tree-LSTM}(\text{Parse}(p)) \tag{4}$$

where $\mathcal{V}_{SE}(p)$ is the output vector. Once $p$ and $h$ are encoded, the final output of this encoder is the concatenation of the two output vectors

$$\mathcal{V}_{SE}(p, h) = \mathcal{V}_{SE}(p) \oplus \mathcal{V}_{SE}(h) \tag{5}$$

### 3.3 Feature Encoder

The explicit integration of entity-level external knowledge has been used to improve many NLP models' performance (Das et al., 2017; Sun et al., 2018). Domain knowledge has also been demonstrated to be useful for in-domain tasks (Romanov and Shivade, 2018; Lu et al., 2019). Therefore, in addition to generic encoders such as MT-DNN and Tree-LSTM, we further enhance the model with domain-specific knowledge through indirectly leveraging labeled biomedical data for other tasks. To do that, we propose a domain feature encoder that identifies and vectorizes biomedical named entities using pre-trained medical taggers and counts (1) the number of each entity type in $p$ and $h$; and (2) the number of shared entities and shared entity types in a $(p, h)$ pair.

In addition to domain knowledge, inspired by Bowman et al. (2015) and Abacha and Demner-Fushman (2016), we also extract generic string features and use them to capture the similarity between $p$ and $h$ and then convert the results into vectors. Such similarity information includes n-gram overlap, Levenshtein distance (Levenshtein, 1966), Jaccard similarity (Jaccard, 1901), ROUGE (Lin, 2004) and BLEU (Papineni et al., 2001) scores, and absolute length difference.[2]

To encode the aforementioned features into vectors, each extracted feature is represented by a single scalar and then grouped with others into an array, denoted by $\mathbf{v}^{(d)}$ and $\mathbf{v}^{(g)}$ for domain and generic features, respectively. Later, they are converted into dense representations by linear transformations and a ReLU nonlinearty. For domain features, this process can be formulated by

$$\mathcal{V}_{FE}^{(d)}(p,h) = \text{ReLU}(\mathbf{W^{(d)}}\mathbf{v}^{(d)} + \mathbf{b}^{(d)}) \qquad (6)$$

and $\mathcal{V}_{FE}^{(g)}(p,h)$ is obtained for generic features in a similar way. As a result, the final output of the feature encoder is the concatenation of vectors with domain and generic knowledge

$$\mathcal{V}_{FE}(p,h) = \mathcal{V}_{FE}^{(d)}(p,h) \oplus \mathcal{V}_{FE}^{(g)}(p,h) \qquad (7)$$

### 3.4 Softmax Classifier

Once the outputs from the aforementioned encoders are obtained, a final representation of $p$ and $h$ is concatenated using the encoded vectors

$$\mathcal{V}(p,h) = \begin{bmatrix} \mathcal{V}_{TE}(p,h) \\ \mathcal{V}_{SE}(p,h) \\ \mathcal{V}_{FE}(p,h) \end{bmatrix} \qquad (8)$$

Then, a softmax classifier is used to compute the class-wise probability distribution from $\mathcal{V}(p, h)$

$$f(p,h) = \text{softmax}(\mathbf{W}\mathcal{V}(p,h) + \mathbf{b}) \qquad (9)$$

Among the three encoders, our base model always includes the text encoder. The other two encoders are optional, leading to different base models, whose performance will be compared in §6.1.

## 4 Model Enhancement

We enhance the base models discussed above with two techniques, namely model ensemble and conflict resolution: ensemble models combine predictions made by different base models, and conflict resolution takes advantage of NLI datasets where multiple $(p, h)$ pairs share the same premise $p$.

### 4.1 Model Ensemble

Model ensemble is a common technique to combine predictions of multiple classifiers for better results (Maclin and Opitz, 1999). In NLI, model ensemble has also been proven helpful (Ghaeini et al., 2018a). In our work, when multiple base models are trained, we follow the strategy in Lee et al. (2015) and Lakshminarayanan et al. (2017) and average the models' predictions by

$$f^{(ME)}(p,h) = \frac{1}{n}\sum_{i=1}^{n} f_i(p,h) \qquad (10)$$

with $n$ denoting the number of ensembled base models and $f_i(p, h)$ being the probability distribution produced by the $i^{\text{th}}$ base model.

### 4.2 Conflict Resolution

Due to the special data collection strategy of MedNLI (see Romanov and Shivade (2018)), each premise is always paired with three hypotheses, each forming an entailment, a neutral, and a contradiction pair with the premise. For example, the premise "Labs were notable for Cr 1.7 (baseline 0.5 per old records) and lactate 2.4." appears three times in the dataset, each pairing with a different hypothesis: (1) "Patient has elevated Cr" (2) "Patient has normal Cr" and (3) "Patient has elevated BUN". The three hypotheses each forms a distinct relationship with the premise. We say the three $(p, h)$ pairs with the same premise form a *group*.

For every group, there are six possible non-conflicting combinations of predictions: $\mathcal{C} = \{\langle\text{e,c,n}\rangle, \langle\text{e,n,c}\rangle, \langle\text{n,e,c}\rangle, \langle\text{n,c,e}\rangle, \langle\text{c,n,e}\rangle, \langle\text{c,e,n}\rangle\}$. Ideally, a model should yield non-conflicting group predictions; that is, $\langle y_{p,h_1}, y_{p,h_2}, y_{p,h_3}\rangle \in \mathcal{C}$ where $h_1$, $h_2$, $h_3$ are the three hypotheses in a group. However, our model determines the label of each pair independently from other pairs in the same group, and thus the three labels could be in conflict. To resolve this conflict, we propose two methods: heuristic processing and multimodal attention. Note that when resolving the conflict,

[2] The choices of metrics are intended to capture a wide range of similarity information, e.g. BLEU for n-gram precision and ROUGE for n-gram recall.

both methods could potentially change the predictions for all three pairs in a group even when only two pairs have conflicting labels.

**Heuristic Processing (HP):** We first use our base or ensemble model to compute the class-wise probability distribution for each $(p, h_i)$ pair

$$f(p, h_i) = \begin{bmatrix} P_e^{(i)} & P_c^{(i)} & P_n^{(i)} \end{bmatrix}^\top \tag{11}$$

where $i \in \{1, 2, 3\}$, and $P_r^{(i)}; r \in \{e, c, n\}$ is the probability of the $i$-th pair having relationship $r$. Then we compute the probability of each non-conflicting combination under this model by

$$P_{\langle r_1, r_2, r_3 \rangle} = \frac{1}{|\mathcal{C}|}(P_{r_1}^{(1)} + P_{r_2}^{(2)} + P_{r_3}^{(3)}) \tag{12}$$

where $\langle r_1, r_2, r_3 \rangle \in \mathcal{C}$.

Finally, we adjust the group predictions taking

$$\langle y_{p,h_1}^{(HP)}, y_{p,h_2}^{(HP)}, y_{p,h_3}^{(HP)} \rangle = \underset{\langle r_1, r_2, r_3 \rangle \in \mathcal{C}}{\arg\max}\ P_{\langle r_1, r_2, r_3 \rangle} \tag{13}$$

Intuitively, for each non-conflicting combination, we add up the prediction probabilities using the model output to derive a combination probability. We take the highest one as the final prediction.

**Multimodal Attention (MA):** We also trained an attention-based neural network to be responsible for conflict resolution so that it can be more expressive at intra-group interactions. It takes the probability distribution from the previous model as well as a positional encoding for input. We added the positional encoding aiming to capture patterns present in the dataset. For each pair, the input of our MA method is

$$\mathbf{p}_i = \begin{bmatrix} P_e^{(i)} & P_c^{(i)} & P_n^{(i)} & i \end{bmatrix}^\top \tag{14}$$

where $i \in \{1, 2, 3\}$ is the index of the pair. We first map it to a hidden space

$$\mathbf{h_i} = \mathbf{W}^{(h)}\mathbf{p}_i + \mathbf{b}^{(h)} \tag{15}$$

We compute intra-group attention by dot-product

$$a_{ij} = \mathbf{h_i} \cdot \mathbf{h_j} \tag{16}$$

Then, we compute attended hidden states by

$$\mathbf{h_i'} = \sum_{j=1}^{3} \frac{\exp(a_{ij})}{\sum_{k=1}^{3} \exp(a_{ik})} \mathbf{h_j} \tag{17}$$

The output probability distribution of $i$-th pair is

$$f^{(MA)}(p, h_i) = \text{softmax}(\mathbf{W}^{(o)}\mathbf{h_i'} + \mathbf{b}^{(o)}) \tag{18}$$

Finally, the prediction is computed by Eq. (2).

| | **Train** | **Dev** | **Test** |
|---|---|---|---|
| # of pairs | 11,232 | 1,395 | 1,422 |
| # of tokens in $p$ | 215k | 29k | 26k |
| # of tokens in $h$ | 66k | 8k | 8k |
| Max. $p$ length | 176 | 110 | 87 |
| Max. $h$ length | 18 | 15 | 16 |
| Avg. $p$ length | 19.2 | 20.4 | 18.6 |
| Avg. $h$ length | 5.8 | 5.7 | 5.7 |

Table 1: Key statistics of the MedNLI dataset. We tokenize the sentences with NLTK (Loper and Bird, 2002).

## 5 Experiment Settings

### 5.1 Data

We use MedNLI as our main training dataset, for it is the official training set of MEDIQA. We also pre-train the text encoder on MIMIC-III discharge summaries (Johnson et al., 2016) using BERT's language modeling objectives (see §3.1).

**MedNLI:** The MedNLI dataset (Romanov and Shivade, 2018) presents unique challenges that require reasoning over biomedical domain knowledge. We use it to train out models and show its statistics in Table 1.

**MIMIC-III:** MIMIC-III (Medical Information Mart for Intensive Care) (Johnson et al., 2016) is a large database with information about patient admission to critical care units. We pre-train on its discharge summaries portion to obtain a better biomedical text representation. After some basic text cleaning, we obtain a corpus with around 7M sentences, 83M words, and 546M characters.

### 5.2 Data Pre-Processing

For pre-processing, we lowercase all our data and use the uncased pre-trained models unless otherwise specified. We replace masked patient health information (PHI) in the form of "[** *text* **]" with pseudo-value generated from gazetteers according to the PHI type[3]. For example, "[** Last Name **]" is replaced with a random last name such as "Smith".

### 5.3 Implementation

For MT-DNN, we use its own hyperparameters without modification. By default, we use 300-dimensional GloVe embeddings trained on Wikipedia and Gigawords (Pennington et al.,

[3] With the tool https://github.com/jtourille/mimic-tools

2014) to initialize the Tree-LSTM, which reduces each parse tree into a 100-dimensional vector. In the feature encoder, we use scispaCy (Neumann et al., 2019) to extract 38 domain features[4]. We also extract 27 linguistic features from the 6 categories specified in §3.3. We project the 38 domain features into $38 \times 20 = 760$ dimensions and the 27 linguistic features into $27 \times 20 = 540$ dimensions with fully-connected layers (See Equation (6)).

We fine-tune the text encoder with MIMIC-III discharge summaries using the same objectives as BERT, i.e. masked language model and next sentence prediction, for 8 epochs.

For training, we use the AdaMax optimizer (Kingma and Ba, 2014) with learning rate $5 \times 10^{-5}$. We use a batch size of 16 and train each model for 15 epochs. All other training hyperparameters are the same as the MT-DNN work.

## 6 Experimental Results

For our experiments, we first find the best configuration for a single base model, and then apply ensemble and conflict resolution on top of it. We run all these experiments with *MT-DNN base* for faster iterations. In order to maximally leverage the MedNLI dataset, unless otherwise specified, all experiments use the MedNLI training and development sets as the training data, and evaluate the performance directly on the MedNLI test set.

After obtaining the best configuration according to the development set performance, we retrain the whole system with that configuration on *MT-DNN large* using the whole MedNLI dataset (i.e. training+development+test). We run it on the MEDIQA Task 1 test set for the final submission (§6.4).

### 6.1 Base Model Results

The base model has many configurations depending on choices of the three encoders, whether to perform language model fine-tuning, and the embedding to use for Tree-LSTM initialization. To find a good, albeit not necessarily optimal, model configuration, we experiment with each modeling decision individually, and greedily use the best option found in the preceding experiments for the ones that follow. We then report ablation results to show the resulting configuration to be a local optimum.

| Text Encoder | SE | FE | Acc. |
|---|---|---|---|
| BERT | | | 79.68 |
| | ✓ | | 79.89 |
| | | ✓ | 79.54 |
| | ✓ | ✓ | 79.96 |
| BioBERT | | | 80.87 |
| | ✓ | | 81.01 |
| | | ✓ | 81.01 |
| | ✓ | ✓ | 81.29 |
| MT-DNN | | | 81.22 |
| | ✓ | | 81.43 |
| | | ✓ | 81.58 |
| | ✓ | ✓ | **81.72** |

Table 2: Performance of the base model with different configurations of the three encoders: text encoder (*TE*), syntax encoder (*SE*), and feature encoder (*FE*). We use GloVe (Embedding I) for Tree-LSTM initialization, and the experiments do not include language model fine-tuning and conflict resolution.

| Pre-Training | Acc. |
|---|---|
| w/o LM fine tuning | 81.72 |
| with LM fine tuning | **83.26** |

Table 3: The effect of pre-training. The first row is the best configuration from Table 2 (MT-DNN + SE + FE + Embedding I). The second row is the same system but pre-trained on MIMIC-III discharge summaries.

**Encoders:** Among the three encoders, the text encoder is the most important, so we will always include it in the base model. We compare three text encoders, including BERT, BioBERT[5] (Lee et al., 2019), and MT-DNN. As for syntax and feature encoders, we compare base models with or without them. The performance of all the combinations are in Table 2, which shows that MT-DNN outperforms BERT and BioBERT, and adding syntax and feature encoders to MT-DNN provides a small improvement[6]. The best result (81.72%) is in the last row and we will refer its configuration as MT-DNN + SE + FE from now on.

**Language Model Fine-Tuning (LMFT):** Using language modeling objective, we fine-tune the text encoder with MIMIC-III discharge summaries. The result is in Table 3, and it demonstrates that the language model fine-tuning scheme

[4] We use scispaCy to identify 18 types of biomedical named entities and turn them into features as mentioned in §3.3. Thus, there are totally $18 \times 2 + 2 = 38$ features.

[5] Because the BioBERT authors only released cased models, we maintain our data casing in relevant experiments.

[6] We also experimented with initializing text encoder word embedding weights with pre-trained static embeddings but it degraded the performance significantly.

| Embedding for Tree-LSTM | Acc. |
|---|---|
| Embedding I | **83.26** |
| Embedding II | 82.91 |
| Embedding III | 82.84 |

Table 4: Effect of different embeddings for Tree-LSTM initialization in the syntax encoder. The first row is the best result from Table 3. The last two rows are the same system but with different embeddings.

brings a significant performance increase. This finding aligns with previous studies (Radford et al., 2019; Devlin et al., 2018; Lee et al., 2019; Alsentzer et al., 2019).

**Syntax Encoder Embeddings:** We used 300-dimensional GloVe embeddings to initialize the Tree-LSTM for Table 2 and 3, and we call it *Embedding I*. Romanov and Shivade (2018) used embeddings trained on biomedical corpora and observed non-trivial accuracy gain over general-domain embeddings. Thus, we also experimented with two domain-specific word embeddings that they used and released to initialize the Tree-LSTM, and we will call them *Embedding II* and *III*. Here is a quick summary of the embeddings:

I. GloVe embedding trained on Wikipedia 2014 + Gigaword 5;

II. Embedding initialized with common crawl[7] GloVe and fine-tuned on BioASQ and then MIMIC-III;

III. Embedding initialized with fastText (Bojanowski et al., 2017) trained on Wikipedia and fine-tuned on MIMIC-III.

Table 4 shows the effect of these embeddings. The first row is the best result from Table 3, which uses Embedding I, and the next two rows are the results when the embedding is changed. The table shows that using specific in-domain embeddings (the second and the third rows in Table 4) does not improve the performance. This is somewhat surprising, but also understandable since these in-domain embeddings are used only in the syntax encoder, instead of being used to initialize the main encoder as in Romanov and Shivade (2018).

**Single Model Ablation:** Table 2-4 show that the best configuration for the base model is MT-DNN + SE + FE + LMFT + Embedding I; that is, it uses all three encoders, is fine-tuned with MIMIC-III discharge summaries, and uses regular GloVe embeddings to initialize the Tree-LSTM.

| Base Model Configuration | Acc. |
|---|---|
| MT-DNN + SE + FE + LMFT + Emb I | **83.26** |
| MT-DNN → BERT | 82.14 |
| MT-DNN → BioBERT | 82.84 |
| – SE | 82.28 |
| – FE | 82.49 |
| – LMFT | 81.72 |
| Emb I → Emb II | 82.91 |
| Emb I → Emb III | 82.84 |

Table 5: The ablation results on top of the best base model. LMFT denotes language model fine tuning.

Because we followed a greedy process for various modeling decisions, there is no guarantee that this configuration is globally or even locally optimal. To test the optimality of the resulting model, we conducted ablations by individually changing each modeling decision on top of the best base model and compare the performance. The results are in Table 5, which show that the greedily found configuration is still the best-performing one among the ablations. In other words, while this configuration is still not guaranteed to be globally optimal, it is at least a locally optimal one.

## 6.2 Model Enhancement Results

We want a diverse set of member models to achieve better ensemble performance. We present ones that lead to better ensemble performance in Table 6. We also report the ensemble models and conflict resolution results in Table 6.

**Ensemble:** With the large number of possible configurations for the base model, it is infeasible to test out all ensemble combinations. On the other hand, the performance of different ensembles does not vary much. We ran all $2^9 - 1 = 511$ ensembles corresponding to all the non-empty subset of the 9 base models A-I, and found that on average on the development set (i.e. original MedNLI test set), ensemble models improve over their best-performing member by $0.86\% \pm 0.51\%$, and over the member average by $1.47\% \pm 0.60\%$. These results demonstrate the general usefulness of the ensemble stage. In Table 6, we show some of the ensemble models, most of which outperform their member models.

[7] https://commoncrawl.org, a corpus that contains 840 billion tokens of web data.

| Model ID | | | | | | | Dev (i.e. MedNLI Test) | | | MEDIQA Test | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (R & S, 2018) | TE | SE | FE | LMFT | Emb | Prepro | Raw | HP | MA | Raw | HP | MA |
| | InferSent | | | | * | | 73.5 | - | - | - | - | - |
| | InferSent | | | | III | | 76.6 | - | - | - | - | - |
| **Base Model** | **TE** | **SE** | **FE** | **LMFT** | **Emb** | **Prepro** | **Raw** | **HP** | **MA** | **Raw** | **HP** | **MA** |
| A | MT-DNN | ✓ | | | I | | 81.36 | 85.16 | 96.20 | 80.49 | 87.16 | 97.28 |
| B | MT-DNN | ✓ | | | II | | 81.50 | 85.94 | 96.62 | 78.77 | 87.41 | 97.53 |
| C | MT-DNN | ✓ | | ✓ | I | | 82.28 | 87.90 | 97.61 | **82.47** | **90.86** | 98.02 |
| D | MT-DNN | ✓ | | ✓ | I | ✓ | 82.49 | 86.36 | 97.75 | **82.47** | 88.64 | 97.53 |
| E | MT-DNN | ✓ | ✓ | ✓ | I | | 82.35 | 86.57 | 97.47 | 80.99 | 88.40 | **99.51** |
| F | MT-DNN | ✓ | ✓ | ✓ | I | ✓ | **83.26** | 87.62 | 98.17 | 81.23 | 86.91 | 97.53 |
| G | MT-DNN | ✓ | ✓ | ✓ | II | ✓ | 82.91 | 86.57 | 97.61 | 81.48 | 89.88 | 98.52 |
| H | MT-DNN | ✓ | ✓ | ✓ | III | ✓ | 82.84 | 86.50 | 97.61 | 80.49 | 89.38 | 98.02 |
| I | BioBERT | ✓ | ✓ | ✓ | I | ✓ | 82.84 | **88.96** | **98.31** | 78.03 | 83.46 | 99.01 |
| **Ensemble** | **Members** | | | | | | **Raw** | **HP** | **MA** | **Raw** | **HP** | **MA** |
| J | A + C + E | | | | | | 83.68 | 88.19 | **98.17** | **83.95** | **93.33** | **99.01** |
| K | A + B + C + E | | | | | | 83.47 | 88.82 | 97.68 | 83.46 | **93.33** | 98.02 |
| L | A + C + D | | | | | | 83.76 | 88.40 | 97.89 | 82.96 | 92.84 | 98.52 |
| M | F + G + H | | | | | | 83.54 | 87.62 | 98.03 | 80.99 | 88.89 | 98.02 |
| N | F + I | | | | | | **83.97** | **89.94** | **98.17** | 82.22 | 88.64 | **99.01** |
| Avg Gain | - | | | | | | - | 4.59 | 14.79 | - | 7.80 | 16.82 |

Table 6: The performance of different ensemble combinations and conflict resolution strategies on our development set (i.e., the original MedNLI test set) and on the MEDIQA shared task test set. All our models in this table (i.e. the *Base Model* and *Ensemble* sections) use MedNLI training and development sets as the training set, while (R & S, 2018) models (Romanov and Shivade, 2018) use only the MedNLI training set for training and MedNLI development set for tuning. The *Prepro* column refers to whether data pre-processing is used (see §5.2). The *Raw*, *HP*, and *MA* columns refer to model performance without and with the two conflict resolution strategies. The results on the MEDIQA test set are computed after the shared task ended and its gold-standard labels were distributed. We report the baseline result and the best extension from Romanov and Shivade (2018) in the first two rows of the table. Their baseline uses the Common Crawl Glove embedding (denoted as *). Note that their results are not directly comparable with ours because they used the MedNLI Test as their test set whereas we use it as our development set. Finally, the last row, *Avg Gain*, is the average gain of HP and MA over Raw when averaged over all the base models and ensembles.

**Conflict Resolution:** We apply heuristic processing (HP) and multimodal attention (MA) to the base models or the ensembles. Both methods improve the performance by large margins.

To our surprise, multimodal attention works much better than heuristic processing, with around 10% absolute difference in accuracy. After a close examination of the training data and the model output, we realize that the MedNLI dataset has a clear label pattern[8] for pairs in the same group (the label sequence being *entailment*, *contradiction*, and *neutral*). Such a pattern is captured by the MA model, but not by the HP one. This finding not only explains the different performance of the two methods, but also reminds us that the high performance of the MA method is largely due to the pattern (or the bias) of this particular dataset.

Taking the best ensemble model N as an example, we study exactly how the two conflict resolution strategies help on the development set. We show relevant statistics in Table 7. As expected, the less conflict there is in a group, the higher the raw accuracy is. We also see that the majority of HP changes are correct for groups with 2 conflicting predictions, but HP does not help groups where all raw predictions are the same. In contrast, because MA takes advantage of the inherent bias of the dataset, all its produced labels are correct. Nevertheless, MA accuracy is still below 100%, because it does not process groups with no conflicts, and raw accuracy on such groups is not at 100%.

### 6.3 Error Analysis

In real use cases, the input to an NLI system is more likely to be standalone $(p, h)$ pairs instead of groups of three $(p, h)$ pairs. Therefore, we conduct error analysis on the output of ensemble systems without conflict resolution. Figure 2 shows

[8] We checked the percentage of groups observing this pattern after the gold standard for test set is released, and it turns out 100% of the groups follow this pattern.

| Conflict Type | # of Groups | # of Pairs | Raw Acc. | HP | | | MA | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | ✗→✓ | ✓→✗ | ✗→✗ | ✗→✓ | ✓→✗ | ✗→✗ |
| 0 | 295 | 885 | 97.06% | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 172 | 516 | 63.57% | 124 | 43 | 8 | 188 | 0 | 0 |
| 3 | 7 | 21 | 33.33% | 7 | 3 | 4 | 14 | 0 | 0 |

Table 7: Conflict resolution results on model N on our development set (i.e., the MedNLI test set). Groups are categorized by **Conflict Type** (i.e., the number of sentence pairs with the same label), which could be 0, 2, or 3. Each group always has three sentence pairs. "Raw Acc." refers to the accuracy without post-processing. For each conflict resolution strategy, we find the $(p, h)$ pairs whose labels are modified by HP or MA, categorize them based on how the updated predictions differ from the raw predictions, and report the number of $(p, h)$ pairs in each category.

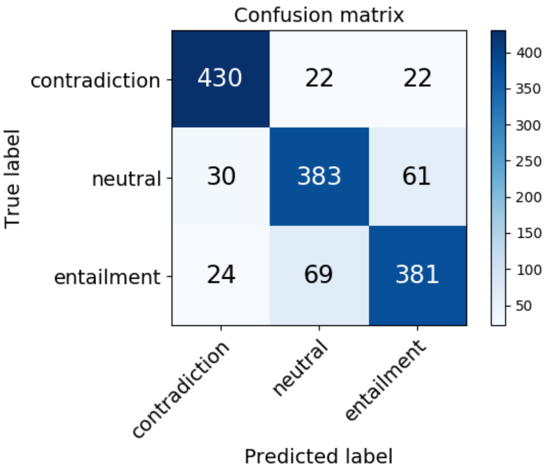

Figure 2: The confusion matrix of Model N before applying conflict resolution strategies.

the confusion matrix for Model N, the best performing ensemble model in Table 6, before conflict resolution. The confusion matrix shows that the model tends to confuse between entailment and neutral. Below are two examples where the model misidentifies entailment pairs to be neutral:

1. ***p***: The patient now presents with metastatic recurrence of squamous cell carcinoma of the right mandible with extensive lymph node involvement.

   ***h***: The patient has oropharyngeal

2. ***p***: In the ED, initial VS revealed T 98.9, HR 73, BP 121/90, RR 15, O2 sat 98% on RA.

   ***h***: The patient is hemodynamically stable.

Both examples contain many medical terms and determining the relationship for the $(p, h)$ pairs is challenging for anyone without medical expertise. Many errors made by the model fall into this category, and fixing them would require the model to be enhanced with deeper domain knowledge.

| Model ID | Conflict Resolution | Acc. |
|---|---|---|
| J | None | 87.2 |
| K | HP | 94.8 |
| L | MA | **98.0** |

Table 8: The results of three models we submitted to MEDIQA Task 1. *Model ID* refers to the model ID in Table 6. The 2nd column denotes different conflict resolution strategies. The *Acc* column is the accuracy on MEDIQA Task 1 test set, which was calculated automatically by the shared task submission site.

### 6.4 Results on MEDIQA Task 1 Test Set

At the time of the shared task submission, we had not completed the systematic experiments as laid out in this paper. We used our then-best ensemble models, re-trained them on *MT-DNN large* using the whole MedNLI set (i.e. training+development+test), and ran them on the MEDIQA Task 1 test set. The results are shown in Table 8. Our best model achieves 98.0% accuracy on the MEDIQA Task 1 test set, the best among all participants.

## 7 Conclusion

We have presented a hybrid architecture for in-domain NLI. Our approach extends current efforts in biomedical NLP (Romanov and Shivade, 2018; Lee et al., 2019) through incorporating auxiliary encoders, domain-specific language model fine-tuning, ensembling, and conflict resolution. We dissected the usefulness of these modeling decisions and provided detailed and systematic ablations. These components work together to form the best performing model on MEDIQA Task 1.

The current system tends to make wrong predictions when in-depth domain-specific knowledge or reasoning is required. For future work, we plan to extend the system to incorporate deeper domain knowledge.

## References

Asma Ben Abacha and Dina Demner-Fushman. 2016. Recognizing question entailment for medical question answering. In *AMIA Annual Symposium Proceedings*, volume 2016, page 310. American Medical Informatics Association.

Asma Ben Abacha and Dina Demner-Fushman. 2017. Nlm_nih at semeval-2017 task 3: from question entailment to question similarity for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 349–352.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *CoRR*, abs/1901.08079.

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. *CoRR*, abs/1904.03323v2.

Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 2019. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the BioNLP 2019 workshop, Florence, Italy, August 1, 2019*. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. *arXiv preprint arXiv:1704.08384*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Reza Ghaeini, Sadid A Hasan, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Z Fern, and Oladimeji Farri. 2018a. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. *arXiv preprint arXiv:1802.05577*.

Reza Ghaeini, Sheik Arick Hasan, Vivek Datla, Joey Liu, Kathy Y. S. Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Z. Fern, and Oladimeji Farri. 2018b. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. In *NAACL-HLT*.

Benjamin Glicksberg, Riccardo Miotto, Kipp Johnson, Shameer Khader, li li, Rong Chen, and Joel T Dudley. 2018. Automated disease cohort selection using word embeddings from electronic health records. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 23:145–156.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Minneapolis, Minnesota, USA*, volume 2.

Paul Jaccard. 1901. Etude comparative de la distribution florale dans une portion des alpes et du jura.

Alistair Edward William Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad M. Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. Mimic-iii, a freely accessible critical care database. In *Scientific data*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. *arXiv preprint arXiv:1806.04330*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. 2015. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.

Miaofeng Liu, Yan Song, Hongbin Zou, and Tong Zhang. 2019a. Reinforced training data selection for domain adaptation. In *Proceedings of ACL, 2019*.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019b. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *HLT-NAACL*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019c. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Mingming Lu, Yu Fang, Fengqi Yan, and Maozhen Li. 2019. Incorporating domain knowledge into natural language inference on clinical texts. *IEEE Access*.

Bill MacCartney and Christopher D Manning. 2009. *Natural language inference*. Stanford University Stanford.

Richard Maclin and David W. Opitz. 1999. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.*, 11:169–198.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*, pages 216–223.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *CoRR*, abs/1902.07669.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *EMNLP*.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning word representations with regularization from prior knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152, Vancouver, Canada.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2*, pages 175–180, New Orleans, Louisiana.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. *arXiv preprint arXiv:1809.00782*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1565–1575.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*

*(Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hongming Zhang, Yan Song, and Yangqiu Song. 2019a. Incorporating Context and External Knowledge for Pronoun Coreference Resolution. In *Proceedings of NAACL-HLT, 2019*.

Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019b. Knowledge-aware Pronoun Coreference Resolution. In *Proceedings of ACL, 2019*.