

Learning Semantic Entailment and Contradiction

Ameera Chowdhury Vaibhav Gandhi Robin Heinonen

UC San Diego

Semantic Entailment

What is semantic entailment?

Semantic entailment is the problem of determining whether one sentence implies another.

A soccer game with multiple males playing.

entailment
E E E E E

Some men are playing a sport.

Definition

If a human reading premise A would infer that hypothesis B is most likely true, then we say that premise A *entails* hypothesis B .

Semantic Contradiction

Definition

We say premise A *contradicts* hypothesis B if a human reading premise A would infer that hypothesis B is most likely false.

A man inspects the uniform of a figure in some East Asian country.

contradiction The man is sleeping
c c c c c

Semantic Neutrality

Definition

The relationship between premise A and hypothesis B is *neutral* if premise A neither entails nor contradicts hypothesis B .

An older and younger man smiling.

neutral
N N E N N

Two men are smiling and laughing at the cats playing on the floor.

Our Task

Determine Relationship between Premise and Hypothesis

Given a premise A and a hypothesis B , we must classify whether

- A entails B ,
- A contradicts B ,
- or whether A and B are neutral.

A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Dataset

Stanford Natural Language Inference Corpus

- 570K pairs of sentences with labels (entailment, contradiction, neutral).
- Sentence pairs and labels written by humans.
- Sentence pairs balanced among labels.
- 57K sentence pairs have four additional judgements for each label.
 - 98% had 3-annotator consensus (gold label).
 - 58% had unanimous consensus.



Baseline model

Features

Following Bowman *et al.*, we extracted:

- BLEU score of hypothesis with respect to premise,
- Length difference between premise and hypothesis,
- Overlap between words in premise and hypothesis as percentage of possible overlap,
- An indicator for every unigram in the hypothesis.

Classifier

We then fit a classifier on top of these features:

- Gaussian Naive Bayes (first three features only),
- Support Vector Machines,
- Logistic Regression (best performance).

Baseline Results: Accuracy Rates

System	Train	Test
Unlexicalized Gaussian Naive Bayes	46.5	47.3
Unlexicalized Support Vector Machine	42.8	46.0
Unlexicalized Logistic Regression	45.9	46.5
Unlexicalized + Unigram Indicator Support Vector Machine	62.9	60.9
Unlexicalized + Unigram Indicator Logistic Regression	66.4	66.4

Baseline Results: Confusion Matrix

Unlexicalized + Unigram Indicator with Logistic Regression

	Contradiction	Entailment	Neutral
Contradiction	2166	573	498
Entailment	464	2479	425
Neutral	684	657	1878

LSTM-based models

- LSTMs [2] ideal for modeling long-term dependencies in sequences
- LSTM-based models outperform handcrafted features on SNLI dataset [1]
- Map the premise and hypothesis to a sequence of 300D vectors using pretrained embeddings from the GloVe algorithm [3]

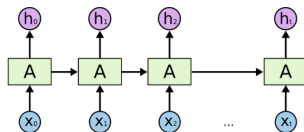


Figure: LSTM-based sequence models

LSTM Model 1

- Process the premise and the hypothesis in succession using the same LSTM
- Pass on the hidden state from premise to the network while processing the hypothesis
- Use MLP to classify based on the hidden representation from the LSTM
- Intuition: Encode the hypothesis in the context of the premise

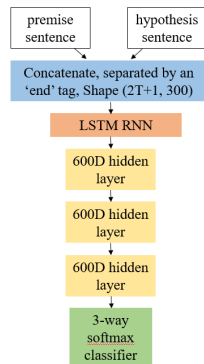


Figure: Pipeline of model.

LSTM Model 2

- Encode the premise and the hypothesis separately
- But tie the weights of their LSTMs
- Concatenate the hidden representations
- Use MLP to classify
- Intuition:
 - Reduce the length of the sequence to be memorized
 - Let the MLP decide how to compare the premise and hypothesis

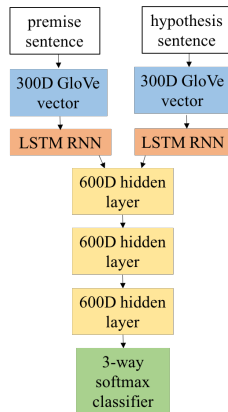


Figure: Pipeline of model.

LSTM Models: More details

- Activation functions: 'relu' and 'tanh'
- Use batch normalization, L2 regularization, and 20% dropout at each layer to improve stability and discourage overfitting
- Optimization Algorithm: RMSprop ($\text{lr} = 0.001$, $\eta = 0.9$)
- Early Stopping: Stop when validation accuracy does not improve for 4 consecutive epochs

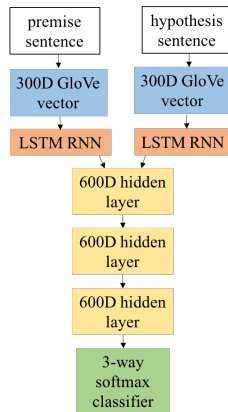


Figure: Pipeline of model.

LSTM model: training/validation errors

Model	Train	Validation
Model 1	65.19%	66.25%
Model 2 w/ hybrid act., shared LSTM	81.62%	76.12%
Model 2 w/ hybrid act., separate LSTMs	78%	74%
Model 2 w/ tanh, shared LSTM	77.2%	74.3%
Model 2 w/ ReLU, shared LSTM	76.1%	73.45%

LSTM model: training/validation errors

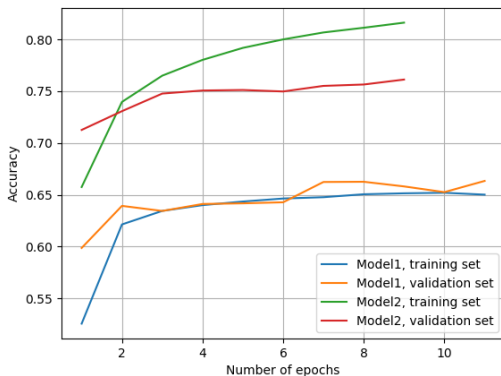


Figure: Training and validation accuracies per training epoch for the two LSTM models.

Best LSTM model: confusion matrix




	Entailment	Contradiction	Neutral
Entailment	2779	168	393
Contradiction	411	2386	475
Neutral	562	336	2332

Table: Confusion matrix for our best LSTM model. Rows are true labels, columns are predictions.

Conclusions

- LSTM-based Deep Neural Networks outperform handcrafted features for the task of NLI when large training datasets are available
- Encoding the premise and the hypothesis separately beats encoding one long concatenated sequence
- Use of different activation functions in different parts of the network can be useful

References I

-  S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning.
A large annotated corpus for learning natural language inference.
CoRR, abs/1508.05326, 2015.
-  S. Hochreiter and J. Schmidhuber.
Long short-term memory.
Neural Computation, 9:1735–80, 12 1997.
-  J. Pennington, R. Socher, and C. D. Manning.
Glove: Global vectors for word representation.
In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

References II



T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kociský,
and P. Blunsom.

Reasoning about entailment with neural attention.

CoRR, [abs/1509.06664](https://arxiv.org/abs/1509.06664), 2015.

Appendix: LSTM-cell structure

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \quad (1)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}^i \mathbf{H} + \mathbf{b}^i) \quad (2)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}^f \mathbf{H} + \mathbf{b}^f) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^o \mathbf{H} + \mathbf{b}^o) \quad (4)$$

$$\mathbf{A}_t = \mathbf{f}_t \circ \mathbf{A}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}^a \mathbf{H} + \mathbf{b}^a) \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t) \quad (6)$$

\mathbf{A}_t are the cells. $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$ are input, forget, and output gates [4].

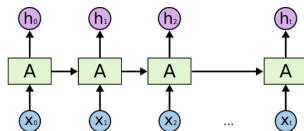


Figure: Basic structure of LSTM (from <http://colah.github.io/posts/2015-08-Understanding-LSTMs>)