CALYPSO: Neural Network Model for Natural Language Inference

Kenny Xu | Colin Man | Kat Gregory

Stanford University | CS224N | March 21, 2017

Introduction

Natural Language Inference (NLI) algorithms solve problems with two main actions: a premise and a hypothesis. Given a **premise**, which is known to be true, and a **hypothesis** whose veracity is unknown, the algorithms determine whether the premise **entails**, **contradicts**, or is **neutral** to the hypothesis. For example, a premise of “some wolves eat deer” implies the hypothesis that “some animals eat deer” since “wolves” is a hyponym of “animals”. However, the same premise would not imply the hypothesis that “some birds eat deer”. The past two years have seen rapid innovation in the NLI field. In particular, Chen et al. ’16 achieved 88.6% accuracy, while Wang ’17 achieved 88.8% accuracy. We introduce CALYPSO, a Neural Network Model for NLI.

Problem Statement

To build on Chen’s EBIM model and explore other approaches to attention and matching inspired by Wang’s Bilateral Multi-Perspective Matching model.

Dataset Processing

We work with the **Stanford Natural Language Inference (SNLI) Corpus**, which contains **570K** English sentence pairs written and labeled by humans. Each premise/hypothesis pair is tagged as either entailment, contradiction, or neutral.

Approach

We first implement a baseline **Bag-of-Words Model** as described Bowman et al. 2015. Next, we implement the *Enhanced BiLSTM Inference Model (EBIM)* as described by Chen ’16, which introduces **soft-attention**. Afterwards, we incorporate three elements of Wang ’17’s *Bilateral Multi-Perspective Matching*: **hard-attention**, **full-matching** and **maxpool-matching**. We then perform an ablation study to determine the importance of the four components.

Method

For our baseline, we use an optimal dropout rate of 10%, a learning rate of 0.001 (Adam), and regularization lambda of .0001, determined through cross-validation. For EBIM and CALYPSO, we use Chen’s optimal parameters with a dropout rate of 50% and a learning rate of 0.0004 (Adam).

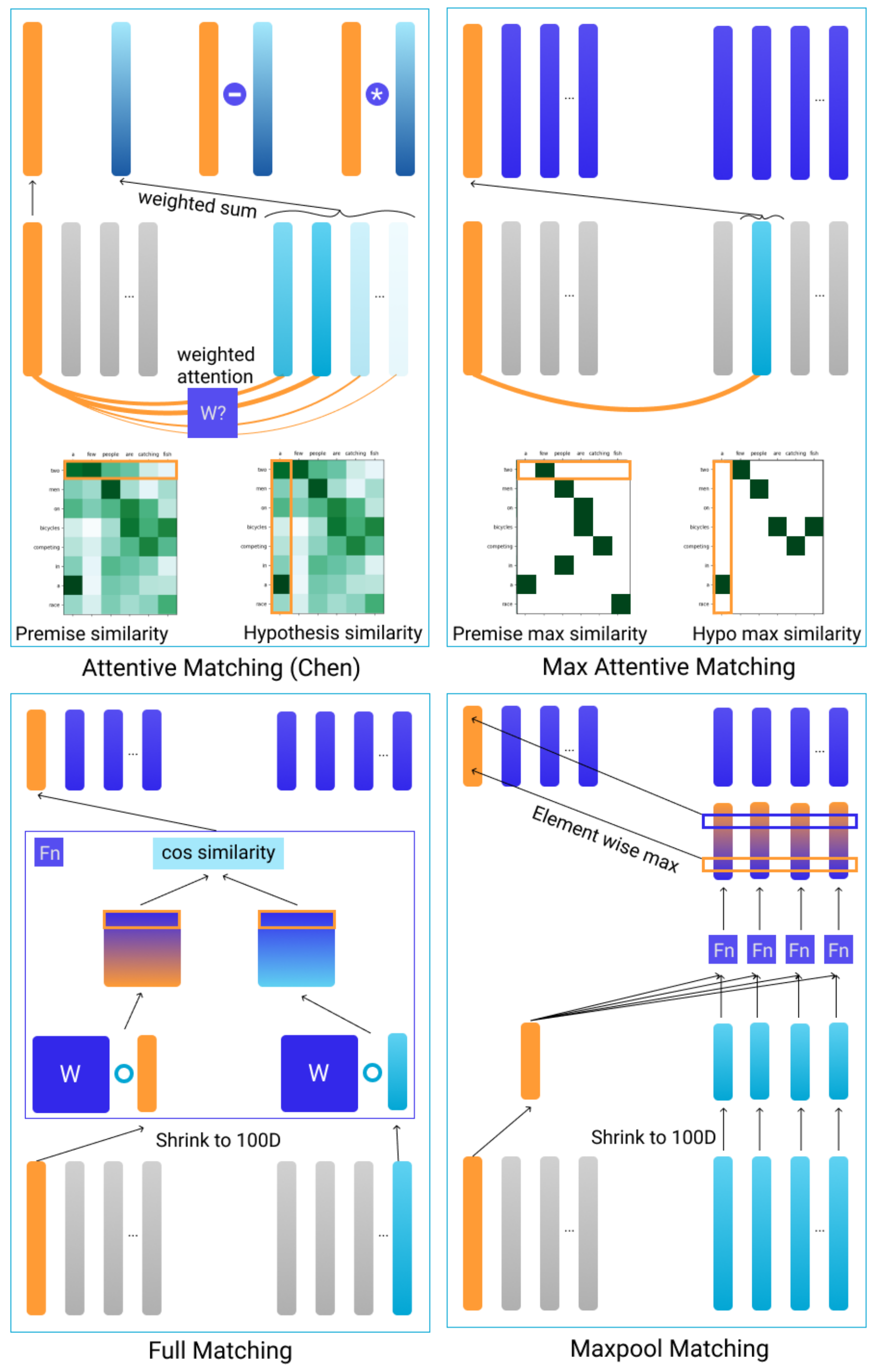


Figure 1: Matching Methods. Attentive soft and hard matching are shown on the top. Multi-perspective "full" and "max" matching are shown at the bottom.

Attention computed between the statements provides alignment context to the composition layer. Matching provides more sophisticated similarity information (20 dimensions for CALYPSO) between words.

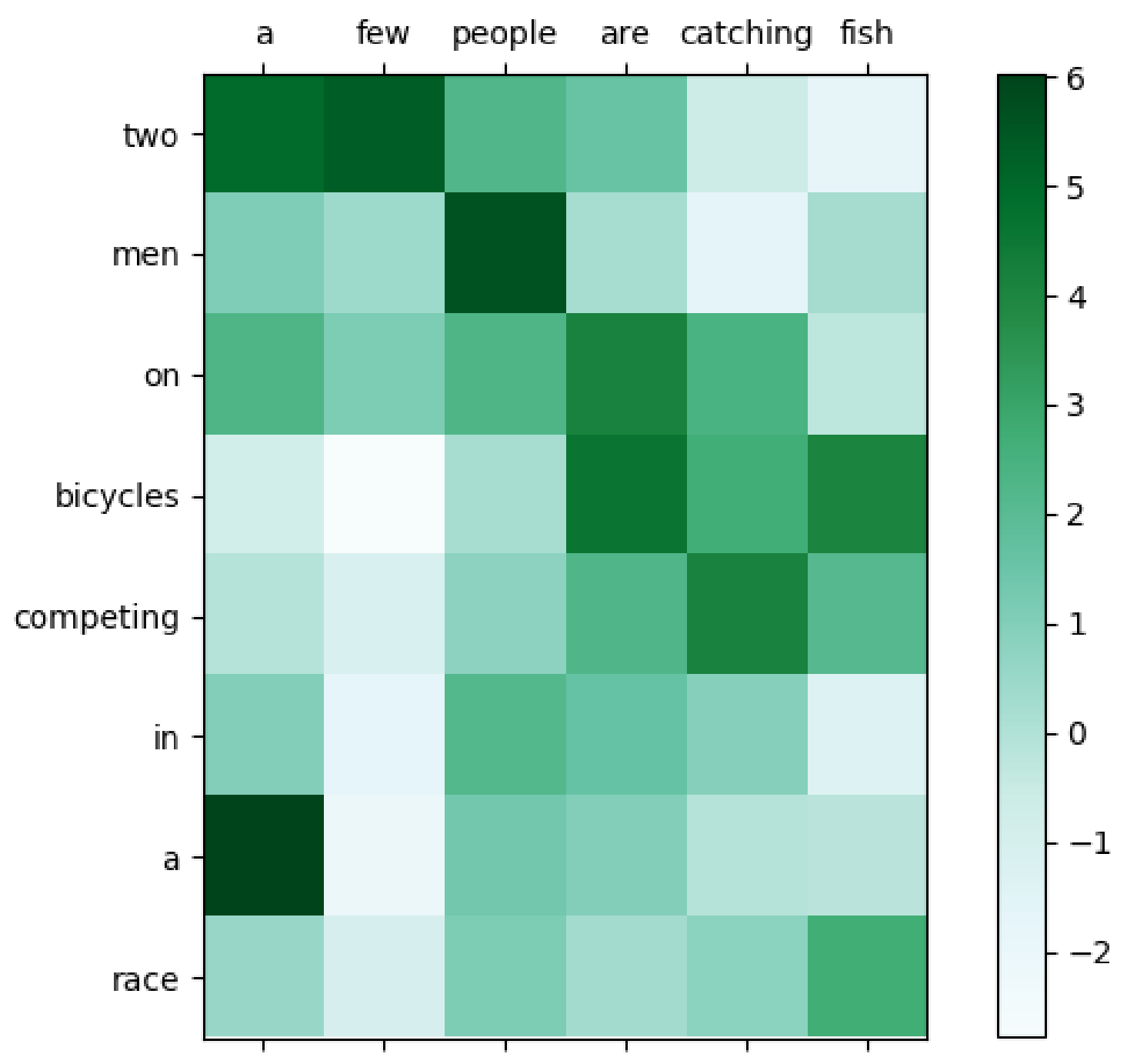
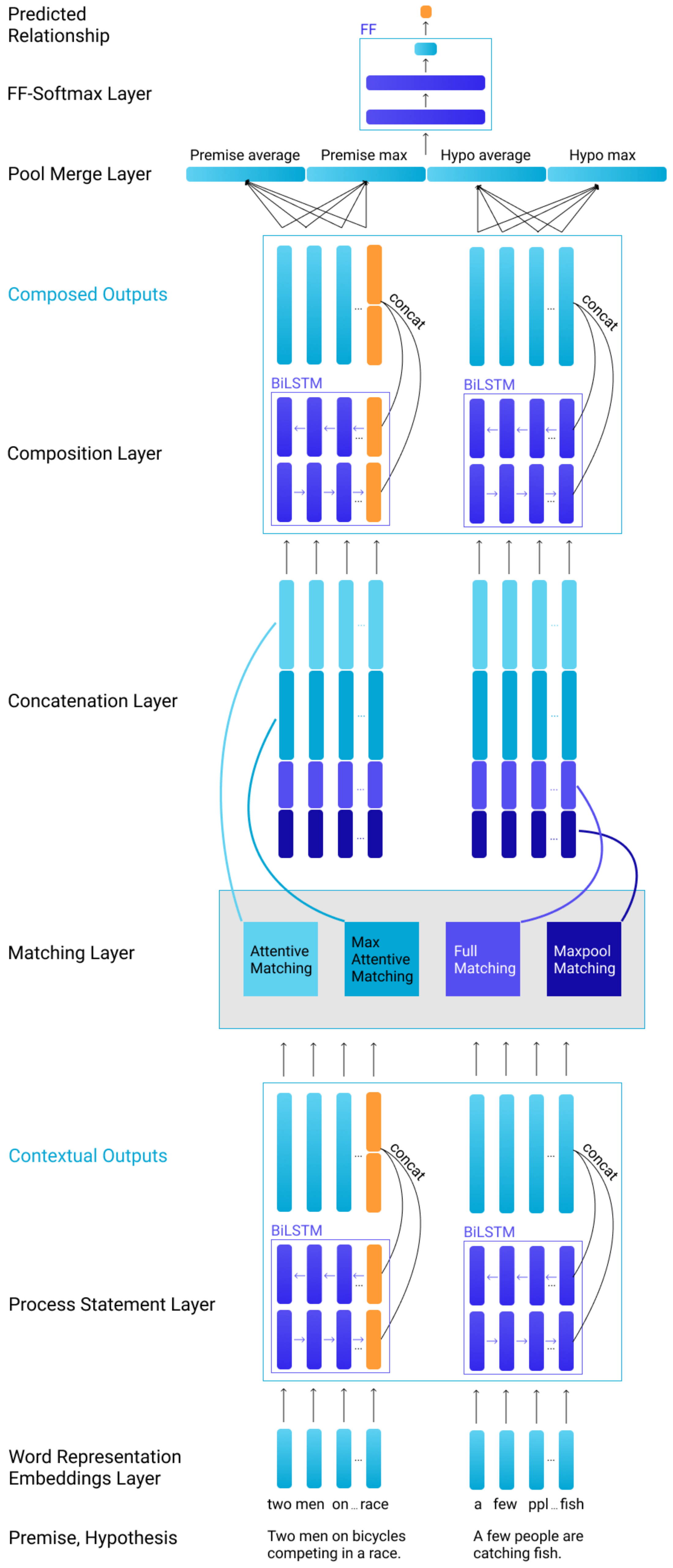


Figure 2: Soft-Attention Matrix Example

Model Architecture



CALYPSO passes word embeddings through the four matching methods shown in Figure 1, concatenates and sends the outputs through a composition layer (BiLSTM), Pool Merge Layer (Avg/Max of Premise and Hypothesis), and 2-layer Feed-Forward Network (tanh), and finally classifies with softmax.

Findings

Model	Train	Dev	Test
CALYPSO	86.6	82.0	**82.0**
w/o wtd. atn.	88.5	**82.2**	81.6
w/o full m.	89.5	81.9	80.7
w/o maxpool m.	**90.2**	81.9	80.9
w/o max atn. m	88.7	81.7	81.0
EBIM + wtd. atn.	93.7	85.5	84.4
EBIM	**93.8**	**86.1**	**85.2**
BOW	78	-	75.6

Figure 3: CALYPSO performance (% accuracy). Weighted (wtd.) attention uses a bilinear product of contextual outputs as weights for matching as opposed to a dot product.

Analysis

EBIM based on Chen ’16 produces the best results across the board. The performance of CALYPSO is 3.2% worse. However, removing any matching method from CALYPSO further reduces our performance. This indicates that each matching method contributes to the accuracy, but that the composition framework needs improvement.

Conclusion

CALYPSO is unable to improve upon EBIM using the matching methods described in Wang ’17. However, each matching method has value, indicating that revision to the composition layer or hyperparameter should improve accuracy.

Future Work

- Find more effective ways of combining the matching methods. Summation, subtraction, and element-wise multiplication are interesting starting points.

- Perform validation to find the optimal learning and dropout rate. We use Chen’s optimal parameters, which are likely suboptimal for CALYPSO given the sizable addition parameters.

- Use distinct underlying LSTMs for attention and full/maxpool matching, allowing each LSTM to specialize.