

ATTENTIVE GUMBEL TREE AND VISUALIZATION IN NATURAL LANGUAGE INFERENCE

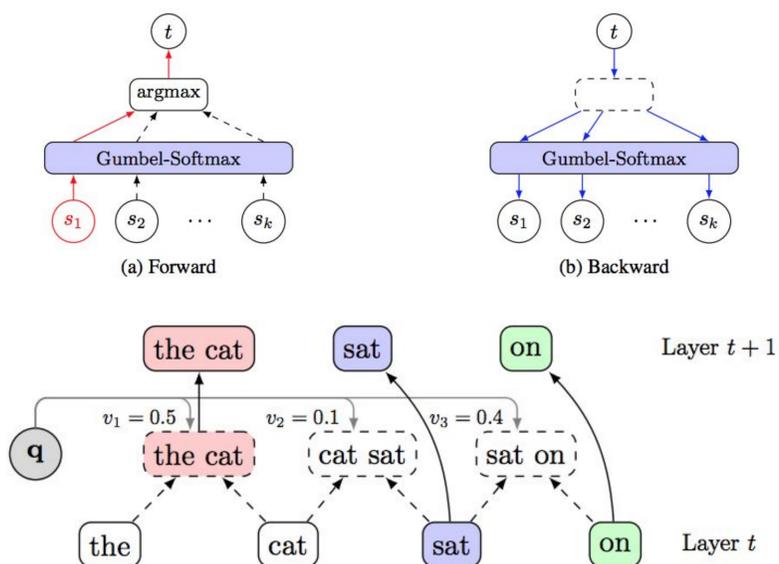
Student: Dinh Ngoc Hai

Supervisor: Assoc Professor Hui Siu Cheung

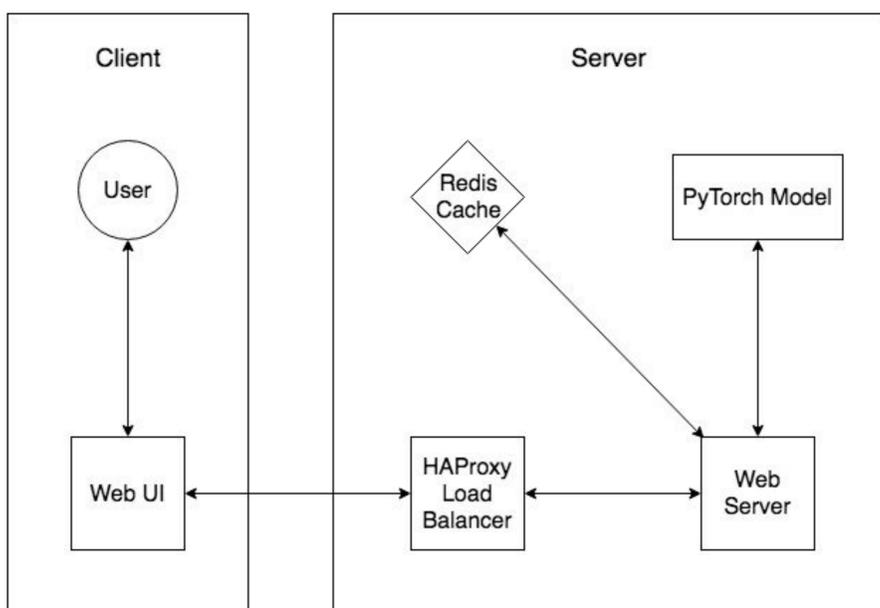
INTRODUCTION

Natural language inference (NLI) is a natural language processing (NLP) task of verifying whether a hypothesis sentence can be inferred directly from a premise sentence. While traditional approaches with separate encoder using recursive neural networks (RvNNs) can achieve high precision, they require structured input and are often unable to show the underlying sentence representation that can be understood by human. This project studies the use of the Gumbel Tree-LSTM, which attempted to construct the task-specific parse-tree representation of sentences only from the plain text data. We will show how the existing model can be improved by introducing an attentive layer and the visualization of the tree structure that can be built by the model for arbitrary sentence. Evaluation of this study is conducted using the Stanford Natural Language Inference (SNLI) Corpus. We also deploy a web prototype system to show the working of the model.

ATTENTIVE GUMBEL TREE LSTM



WEB SYSTEM ARCHITECTURE



DATASET

Premise sentence	Hypothesis sentence	Verdict
Three bikers stop in town	Three bikers are going 100 mph	Contradiction
Wet brown dog swims towards camera	The dog is sleeping in his bed	Contradiction
Children smiling and waving at camera	There are children present	Entailment
A man walking proudly down the street	A man is outside	Entailment
Two small dogs run across the green grass	Some small dogs are running to their owners	Neutral
A man reads the paper in a bar with green lighting	The man is reading the sportspage	Neutral

RESULTS

Model	Time per epoch (hours)	Accuracy
Baseline Gumbel Tree-LSTM	0.65	82.4%
Gumbel Tree-LSTM with attention layer after hypothesis tree	0.65	82.6%
Gumbel Tree-LSTM with layer-by-layer attention	0.70	84.3%
Gumbel Tree-LSTM with both attention layer after and layer-by-layer attention	0.70	84.0%

WEB SYSTEM INTERFACE