

# A Brief Introduction to Me and My Research

Yuling Shi

SUFE

April, 2021

# Contents

## About Me

## Research Projects

Question Answering (Feb 2020 - Jun 2020)

Kaggle Sentence Classification (Nov 2020 - Dec 2020)

Finite Element Method (Jul 2020 - Feb 2021)

Interpreting NLP Model (Jan 2021 - Present)

## Summary

## About Me

- ▶ Junior from Shanghai University of Finance and Economics majoring in Applied Mathematics (Elite Program), Major GPA (3.64/4)
- ▶ Boardly interested in deep learning, machine learning and scientific computing.
- ▶ Have done many researches about NLP. One earlier paper accepted about scientific computing.

# Outline

About Me

Research Projects

Question Answering (Feb 2020 - Jun 2020)

Kaggle Sentence Classification (Nov 2020 - Dec 2020)

Finite Element Method (Jul 2020 - Feb 2021)

Interpreting NLP Model (Jan 2021 - Present)

Summary

# Background

- ▶ Leader of the project. Taking DL class with juniors from Elite Program in Department of Electrical Engineering in my second year.

## Data description

- ▶ The Natural Questions (NQ) (**Kwiatkowski et al., 2019**) dataset from Google AI.
- ▶ Each example comprised of a google query and a corresponding Wikipedia page.

**Question:** why does queen elizabeth sign her name elizabeth r

**Wikipedia Page:** Royal\_sign-manual

**Long answer:** The royal sign-manual usually consists of the sovereign's regnal name (without number, if otherwise used), followed by the letter R for Rex (King) or Regina (Queen). Thus, the signs-manual of both Elizabeth I and Elizabeth II read Elizabeth R. When the British monarch was also Emperor or Empress of India, the sign manual ended with R I, for Rex Imperator or Regina Imperatrix (King-Emperor/Queen-Empress).

**Short answer:** NULL

**Figure:** Example annotations from the corpus

# Key Experiments

- ▶ Fine-tuning on SQuAD 2.0
- ▶ Mixed Precision Training
- ▶ Hard Negative Sampling
- ▶ Sifting candidates

# Fine-tuning on SQuAD 2.0

- ▶ SQuAD 2.0 - 130,000 crowd sourced question and answer training pairs derived from Wikipedia paragraphs.

## Performance in Training

Model	Start_acc	End_acc	Class_acc
BERT base (Original)	59.1	61.5	73.9
ALBERT xlarge (Original)	0.12	0.13	66.67
ALBERT xlarge (Finetuned)	82.54	86.37	85.75

batch size = 3 per GPU, learning rate = 1e-5 for 3 epochs

# Mixed Precision Training

- ▶ To save memory and speed up - only had 1080Ti GPUs cluster.

## Experiment Result

Precision	EM	F1	Speed up*
FP32 only	84.86	88.00	1.0
FP16 Only	16.75	17.35	1.35x
Mixed precision	84.94	87.97	1.05x

\*All tested during SQuAD2.0 fine-tuning

# Hard Negative Sampling

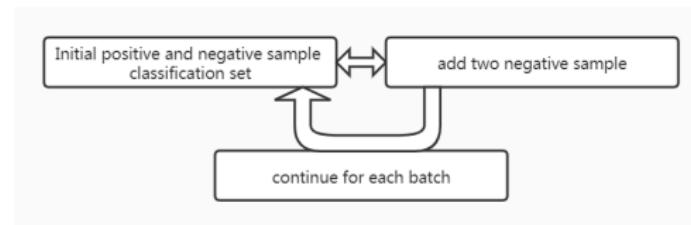
## Intuition

- ▶ Training questions without any short answer(65%) → Too many negative Examples
- ▶ Uniform sampling → most of the negative candidates are "**too easy**"
- ▶ Hard negative sampling → increase the difficulty of training

# Hard Negative Sampling

## Procedure

1. Train a model with uniform sampling and predict on the whole training data
2. Store the answer probability for each negative candidate
3. Normalize the probabilities within documents to form a distribution
4. Sample negative candidates from the probability distribution in training.



# Hard Negative Sampling

## Result

- ▶ Result: the performance was improved 6.6% on Public leaderboard and 9.8% on Private leaderboard.

Model	Public F1	Private F1
BERT baseline	0.516	0.482
BERT with Hard Negative Sampling	0.579	0.574

# Sifting candidates

## Sifting candidates with BERT base to reduce candidates

1. First perform a full prediction on the validation set using a fast model (BERT Base) to reduce number of candidates.
2. Then use larger model to make predictions on the selected candidates.

## Benefits

1. Reduce much predicting time when adding large models.
2. More convenient to ensemble other models.

# Sifting candidates

## Performances

Model	Public F1	Private F1
BERT baseline	0.516	0.482
BERT Base (Hard Negative Sampling)	0.579	0.574
BERT* Sifted → ALBERT xlarge	0.640	0.659

\* also trained with hard negative sampling

# Final Result

Model	Public F1	Private F1
Kaggle Best	0.713	0.717
BERT Base baseline	0.516	0.482
BERT Base (Hard Negative Sampling)	0.579	0.574
BERT Sifted* → ALBERT xlarge	<b>0.640</b>	<b>0.659</b>
Sifted → BERT Base+ALBERT <sub>(ensemble)</sub>	0.665	0.666
Sifted → BERT Large+ALBERT <sub>(ensemble)</sub>	<b>0.738</b>	<b>0.718</b>

\* Sifted here stands for first using BERT base to sift candidates

# Conclusion

- ▶ Learned to search and read latest paper regularly and looked for many useful techniques.
- ▶ Comfortable with Linux environment and commands, and also implementation of DL models.
- ▶ Collaborated with group to discuss and do experiments together. Also asked for teachers' advice regularly.

# Outline

About Me

## Research Projects

Question Answering (Feb 2020 - Jun 2020)

Kaggle Sentence Classification (Nov 2020 - Dec 2020)

Finite Element Method (Jul 2020 - Feb 2021)

Interpreting NLP Model (Jan 2021 - Present)

Summary

## Overview

- ▶ Data: complaints from citizens. The task is to predict label (200 classes in total).
- ▶ Pre-processed the dataset, pre-trained models on similar dataset THUCNews with mixed precision training.
- ▶ Focal loss to track difficult and rare class examples.
- ▶ Designed a auxiliary sentence pair task.
- ▶ Also tried adversarial training, data augmentation, adding other layers after BERT, using RoBERTa-wwm, ERNIE, etc.

## Focal Loss

- ▶ The 200 labels are long-tailed distributed.
- ▶ Scaled losses according to how difficult the example is to predict.

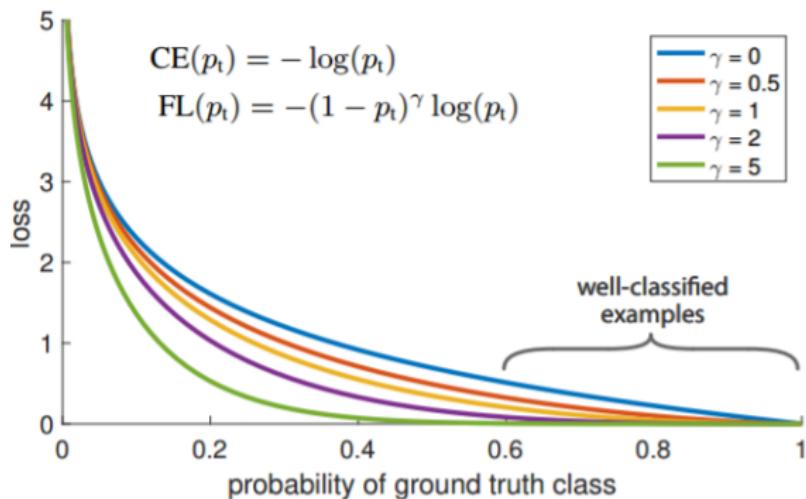


Figure: Different Loss Functions

## Auxiliary Task

- ▶ The original classification task failed to utilize information in the labels.
- ▶ Sort predicted labels by probabilities, 79% of the true label are at the first place, 11% at the second, 96% of them are within top 5 predictions.

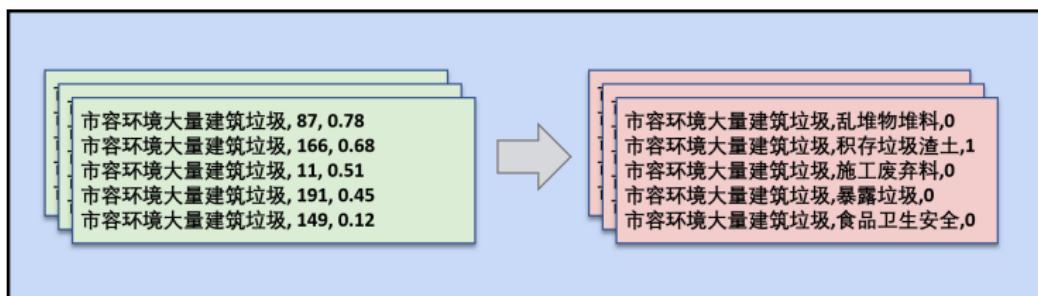


Figure: Generating pair data

# Final Submission

- ▶ Highest public score in class (led by 0.4%) but turned out to be overfitted. (why? lack of cross validation?)

Table: Selected Experiment Results

Model	Public score	Private score
ERNIE <sup>1</sup>	0.7981	0.8000
ERNIE <sup>2</sup>	0.7995	0.8030
ERNIE <sup>3</sup>	0.8049	0.8010

---

<sup>1</sup>Original Task: text classification

<sup>2</sup>Focal Loss

<sup>3</sup>Auxiliary Task: sentence pair classification

# Conclusion

- ▶ Explored and implemented more useful techniques myself.
- ▶ Faced with a more unpredictable project, designed an auxiliary task which performed "well".
- ▶ Experimented more failed techniques and analyzed possible reasons.

# Outline

## About Me

## Research Projects

Question Answering (Feb 2020 - Jun 2020)

Kaggle Sentence Classification (Nov 2020 - Dec 2020)

**Finite Element Method (Jul 2020 - Feb 2021)**

Interpreting NLP Model (Jan 2021 - Present)

## Summary

# Biref Introduction

Finite element space:

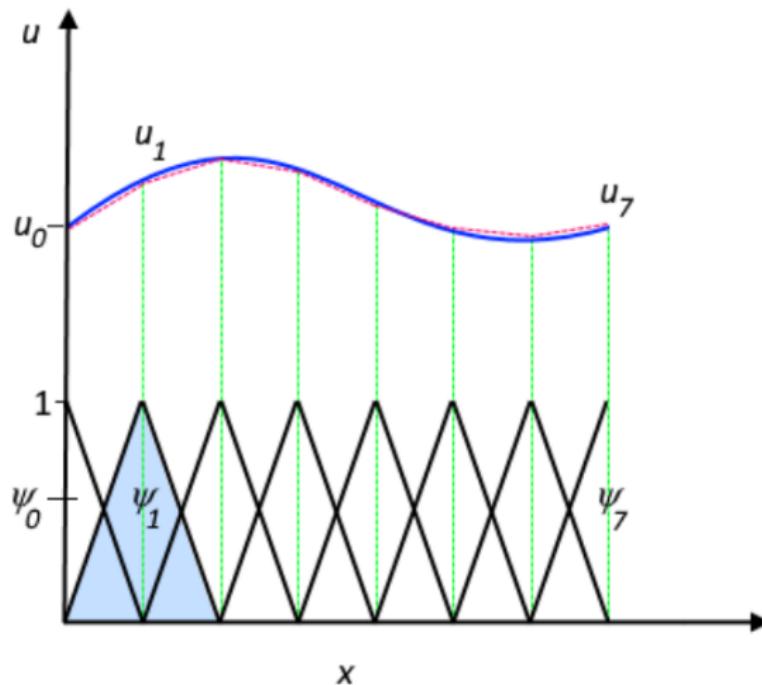


Figure: Linear basis in 1D

# Main Problem

Equation:

$$\begin{cases} \varepsilon^2 \Delta^2 u - \Delta u = f & \text{in } \Omega, \\ u = \partial_n u = 0 & \text{on } \partial\Omega, \end{cases}$$

- ▶ Original variational form

$$\varepsilon^2 (\nabla_h^2 u_{h0}, \nabla_h^2 v_h) + (\nabla_h u_{h0}, \nabla_h v_h) = (f, P_h v_h) \quad \forall v_h \in V_{h0}.$$

- ▶ Original ways to solve:

- ▶ Conforming elements: computational expensive
- ▶ Non-conforming elements: isn't convergent

## Biref Description

Our work:

- ▶ Modified the right hand side via projection:

$$(\nabla w_h, \nabla \chi_h) = (f, \chi_h) \quad \forall \chi_h \in W_h$$

$$\varepsilon^2 a_h(u_{h0}, v_h) + b_h(u_{h0}, v_h) = (\nabla w_h, \nabla_h v_h) \quad \forall v_h \in V_{h0}$$

- ▶ Decoupled the left hand side into four simple equations:

$$(\operatorname{curl}_h z_h, \operatorname{curl}_h v_h) = (\nabla w_h, \nabla_h v_h) \quad \forall v_h \in V_{h0}$$

$$(\phi_h, \psi_h) + \varepsilon^2 (\nabla_h \phi_h, \nabla_h \psi_h) + (\operatorname{div}_h \psi_h, p_h) = (\operatorname{curl}_h z_h, \psi_h) \quad \forall \psi_h \in V_{h0}^{CR}$$

$$(\operatorname{div}_h \phi_h, q_h) = 0 \quad \forall q_h \in Q_h$$

$$(\operatorname{curl}_h u_{h0}, \operatorname{curl}_h \chi_h) = (\phi_h, \operatorname{curl}_h \chi_h) \quad \forall \chi_h \in V_{h0}$$

## Biref Description

- ▶ Can be solved efficiently with the simplest Morley element.

$h$	#dofs	Eq.(5.1)	Eq.(5.7a)	Eq.(5.7b)-(5.7c)	Eq.(5.7d)
		steps	steps	steps	steps
$2^{-1}$	24	1	1	16	1
$2^{-2}$	112	1	4	27	3
$2^{-3}$	480	4	5	34	5
$2^{-4}$	1984	6	7	34	7
$2^{-5}$	8064	6	9	41	9
$2^{-6}$	32512	7	11	43	11
$2^{-7}$	130560	7	14	44	14
$2^{-8}$	523264	9	17	46	17
$2^{-9}$	2095104	9	20	50	21
$2^{-10}$	8384512	12	27	55	27

Figure: Robust iteration steps when solving

# Conclusion

- ▶ Final paper accepted by *Journal of Scientific Computing*
- ▶ Found a suitable open source package myself and studied many bottom-level codes.
- ▶ Fixed many bugs in developing experiments by discussing. Reported bug for the package and contributed codes to develop it.

# Outline

About Me

## Research Projects

Question Answering (Feb 2020 - Jun 2020)

Kaggle Sentence Classification (Nov 2020 - Dec 2020)

Finite Element Method (Jul 2020 - Feb 2021)

Interpreting NLP Model (Jan 2021 - Present)

Summary

# Background

## Existing methods

- ▶ Gradient based methods: gradient, dot product with embeddings, integrated gradient ...
- ▶ Perturbation based methods: input reduction, adversarial perturbations ...

Trying to explain what the model is learning during the training process.

$$\begin{aligned} \text{Loss}(x_1, \dots, x_d) &= \sum_{n_1=0}^{\infty} \dots \sum_{n_d=0}^{\infty} \frac{(x_1 - a_1)^{n_1} \dots (x_d - a_d)^{n_d}}{n_1! \dots n_d!} \left( \frac{\partial^{n_1+\dots+n_d} f}{\partial x_1^{n_1} \dots \partial x_d^{n_d}} \right) (a_1, \dots, a_d) \\ &= f(a_1, \dots, a_d) + \sum_{j=1}^d \frac{\partial f(a_1, \dots, a_d)}{\partial x_j} (x_j - a_j) + \frac{1}{2!} \sum_{j=1}^d \sum_{k=1}^d \frac{\partial^2 f(a_1, \dots, a_d)}{\partial x_j \partial x_k} (x_j - a_j) (x_k - a_k) \\ &\quad + \frac{1}{3!} \sum_{j=1}^d \sum_{k=1}^d \sum_{l=1}^d \frac{\partial^3 f(a_1, \dots, a_d)}{\partial x_j \partial x_k \partial x_l} (x_j - a_j) (x_k - a_k) (x_l - a_l) + \dots \end{aligned}$$

# Dataset

- ▶ e-SNLI dataset: essential words are highlighted by annotators

---

Premise: An adult dressed in black **holds a stick**.

Hypothesis: An adult is walking away, **empty-handed**.

Label: contradiction

Explanation: Holds a stick implies using hands so it is not empty-handed.

---

Premise: A child in a yellow plastic safety swing is laughing as a dark-haired woman in pink and coral pants stands behind her.

Hypothesis: A young **mother** is playing with her **daughter** in a swing.

Label: neutral

Explanation: Child does not imply daughter and woman does not imply mother.

---

Premise: A **man** in an orange vest **leans over a pickup truck**.

Hypothesis: A man is **touching** a truck.

Label: entailment

Explanation: Man leans over a pickup truck implies that he is touching it.

---

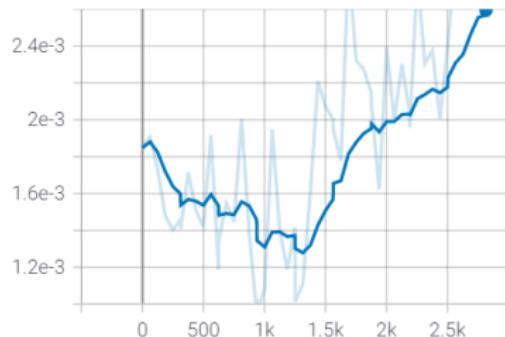
Figure: Examples from e-SNLI

# Experiments

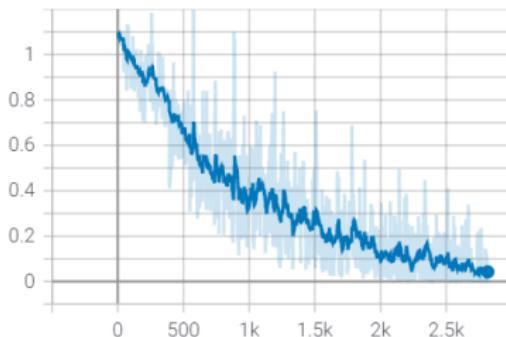
- ▶ Loss during training BERT-Base:

Loss

test  
tag: Loss/test



train  
tag: Loss/train

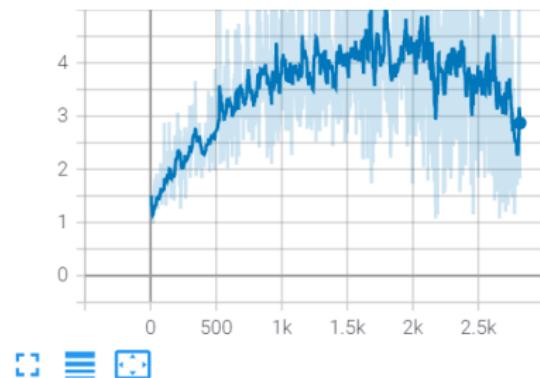


# Experiments

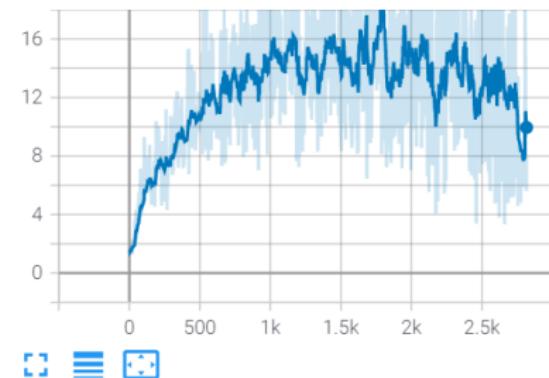
- ▶ Gradients of annotated "essential words" during training  
BERT-Base:

Grad\_loss

grad0\_loss  
tag: Grad\_loss/grad0\_loss



grad\_loss  
tag: Grad\_loss/grad\_loss

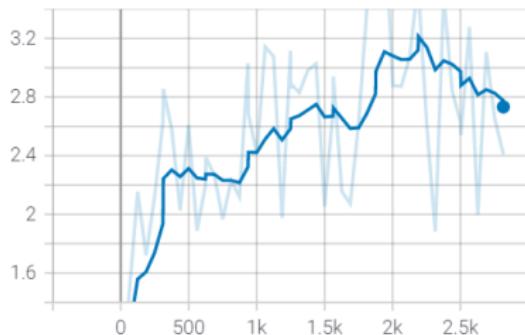


# Experiments

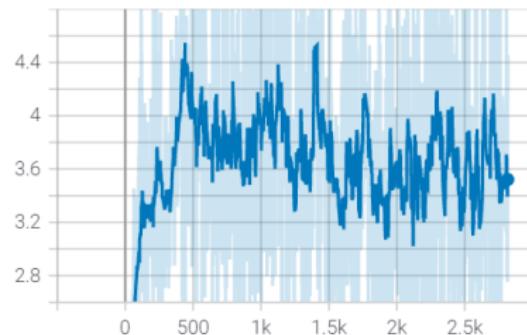
- ▶ Ratio:  $\frac{|\text{Gradients of annotated}|}{|\text{Gradients of all}|}$  during training BERT-Base:

Ratio

test  
tag: Ratio/test

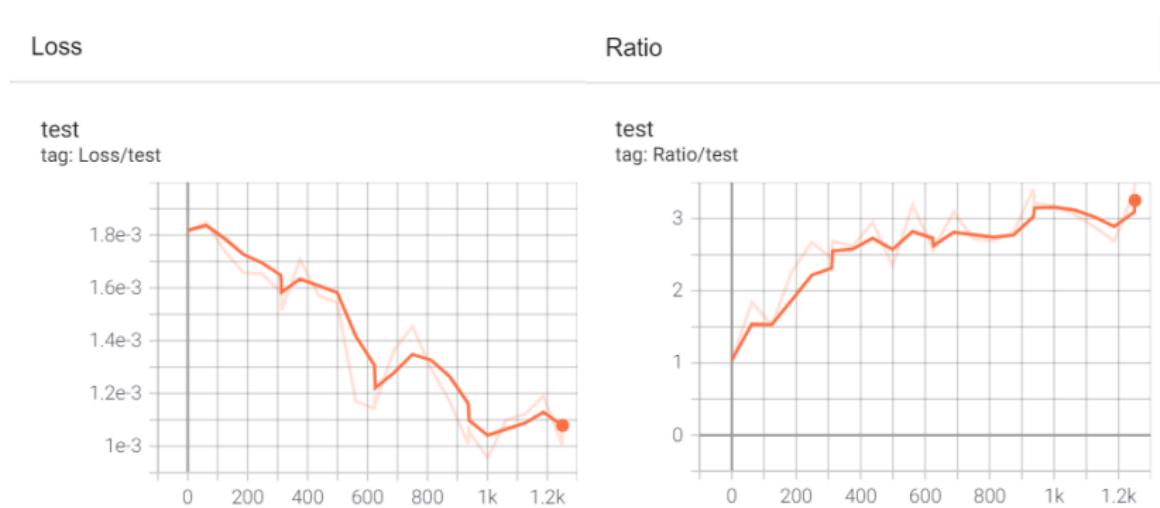


train  
tag: Ratio/train



# Experiments

- ▶ Model learning relations between "essential" words?
- ▶  $\frac{\partial Loss}{\partial E_i E_j}$  during training BERT-Base:



# Experiments

- ▶ What relations are captured?
- ▶ Sentences (label: entailment):
  1. Two young boys of opposing teams play football, while wearing full protection uniforms and helmets.
  2. Boys play football.
- ▶ 'essential words':  
['boys', 'opposing', 'teams', 'play', 'boys', 'play', 'football']

# Experiments

## ► Sentences:

1. Two young boys of opposing teams play football, while wearing full protection uniforms and helmets.
2. Boys play football.

		word	most relavent	score
0		[SEP]	[SEP]	0.044089
2		[CLS]	[SEP]	0.025706
4		[CLS]	[SEP]	0.024833
6		[SEP]	.	0.014351
8		football11	[SEP]	0.014056
10		.	[SEP]	0.013824
12		[SEP]	football12	0.011118
14		[SEP]	football11	0.010987
16		football12	[SEP]	0.010924
18		[SEP]	helmets	0.010910

Figure: At the beginning of training

# Experiments

## ► Sentences:

1. Two young boys of opposing teams play football, while wearing full protection uniforms and helmets.
2. Boys play football.

	word	most relavent	score
0	football11	football12	0.066224
2	and	football12	0.053644
4	football12	protection	0.034782
6	boys	football12	0.025395
8	football12	play	0.021575
10	uniforms	football12	0.018982
12	[CLS]	football12	0.018318
14	[SEP]	football12	0.013509
16	helmets	football12	0.012761
18	football11	protection	0.012189

Figure: Trained for 1/2 epoch

# Experiments

## ► Sentences:

1. Two young boys of opposing teams play football, while wearing full protection uniforms and helmets.
2. Boys play football.

	word	most relavent	score
0	helmets	football12	0.104266
2	football12	football11	0.101465
4	football12	uniforms	0.090533
6	football11	helmets	0.085855
8	helmets	uniforms	0.063052
10	uniforms	[CLS]	0.048258
12	football12	play	0.046607
14	helmets	protection	0.044967
16	football12	[SEP]	0.042081
18	play	helmets	0.038749

Figure: Trained for 1 epoch

## Future Work

- ▶ Search for other dataset to simplify the problem.
- ▶ Techniques to reduce computational cost (randomized?).
- ▶ Design other experiments to figure out what is decreasing?

# Summary

- ▶ Comfortable with math and can implement experiments fast.
- ▶ Highly motivated and always passionate to search for and think of more techniques.
- ▶ Seeking for further guidance :).

*Thanks for your attention!*