NL2code

Sai Krishna Karanam karanam.s@husky.neu.edu
Nischal Mahaveer Chand mahaveerchand.n@husky.neu.edu
Varun Sundar Rabindranath rabindranath.v@husky.neu.edu

I. Introduction

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

II. Related Word

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

III. Dataset

A. Standard Datasets

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

1) Django: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2) HS: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

B. Our dataset

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

IV. Models

We describe four models, each is build upon Pengcheng's model, but encodes the input in a different method. First we start by briefly describing Pengcheng's model.

A. Pengcheng's model

Pengcheng recognized that adding syntax information to the model would give better predictions results XXX. His model, follows an encoder-decoder architecture, and takes the raw comment as input and generates an Abstract Syntax Tree (AST) of the corresponding code as output.

The encoder comprises of an embedding layer and a Bidirectional LSTM (BiLSTM) layer. It takes a comment as input, embeds each word in the comment to give token embeddings $\mathrm{TE_t}$, for each word t in the comment. Each $\mathrm{TE_t}$ is sequentially feed into the BiLSTM layer, to produces a Query Embedding (QE) of 128 dimensions. QE is passed to the decoder module.

The decoder uses an Recurrent Neural Network (RNN)

to sequentially generate each node of the AST. Each node maps to a timestep in the RNN decoding process and thus, generating the AST can be interpreted as unrolling the RNN. The RNN also maintains an internal state to track the generation process at each timestep.

The decoder is slightly complicated and works in mysterious ways! Lord Voldomort himself blessed it with his divine wand to produce a magical black box, that generates ASTs!

Pengcheng's tackles the problem with a probabilistic grammar model of generating an AST y given NL description x: $p(y|x)$. The best possible AST $\hat{y}$ is given by:

$$\hat{y} = \arg\max_{y} p(y|x) \tag{1}$$

$\hat{y}$ is then deterministically converted to the corresponding Python code using astor XXX.

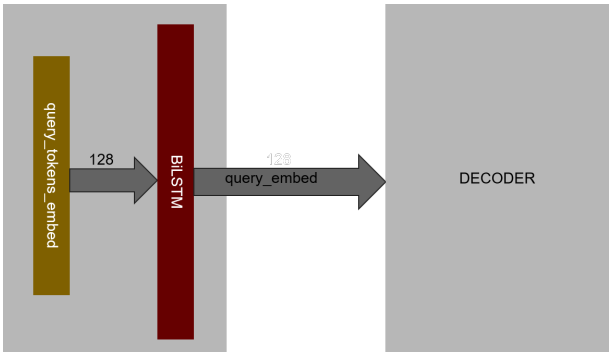

Figure 1: Architecture of Pengcheng's model.

B. Our models

As described, the decoder is already state-of-the-art XXX, and needs no further modifications. All models described hereon use the same decoder architecture as Pengcheng's model with modifications to the encoder and the input data XXX. All models are trained and tested using the dataset decribed in SECTION.

1) Basic Concat (BC): For our first attempt to incorporate syntax information into the encoder, we decided to concatenate (shown as ":") the POS and phrase ID of each token XXX to the corresponding token embedding, giving us the Augmented Token Embedding (ATE). The ATE is then feed into a modified BiLSTM layer that takes 130 dimension embeddings, rather than the default 128 dimensions.

TE_t dimensions: 128
POS_ID_t and $\text{Phrase\_ID}_\text{t}$ dimensions: 1 each; total 2
ATE_t dimension = 130

$\text{ATE}_\text{t} = [\text{TE}_\text{t} : \text{POS\_ID}_\text{t} : \text{Phrase\_ID}_\text{t}]$
QE = BiLSTM(ATE)

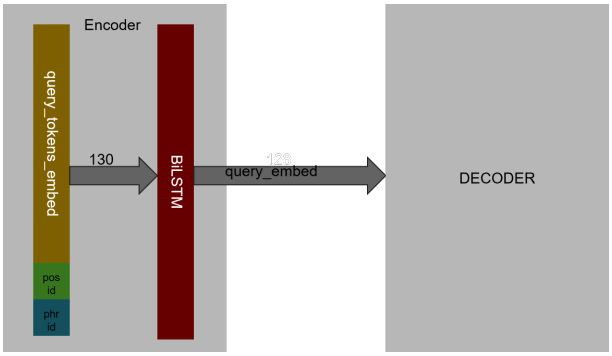

Figure 2: Architecture of Pengcheng's model.

2) Linear Projection (LP): To add some syntactic information over a sequence of tokens, we used an embedding layer for POS and phrase tags. The resulting TE, POS embedding (POSE), and Phrase embedding (PhE) are then concatenated to produce the ATE; which is a $(128 * 3)$ dimension vector. We then apply a linear projection (using a dense layer with the linear activation function) to give ATE Projected (ATEP) which is passed to BiLSTM.

TE_t dimension: 128
POSE_t dimension: 128
PhE_t dimension: 128
Dense = (128 * 3) input nodes, 128 output nodes

$\text{ATE}_\text{t} = [\text{TE}_\text{t} : \text{POSE}_\text{t} : \text{PhE}_\text{t}]$
$\text{ATEP}_\text{t} = \text{Dense}(\text{ATE}_\text{t})$
QE = BiLSTM(ATEP)

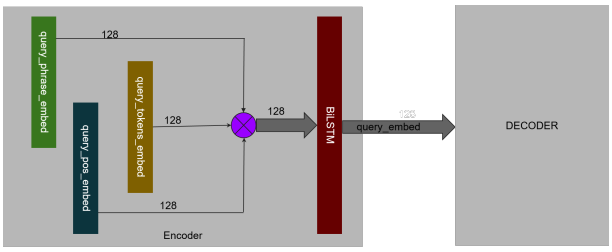

Figure 3: Architecture of Pengcheng's model.

3) Linear Projection Reduced Dimension (LPrd): Subsequently, we noticed that the POS and Phrase vocabulary sizes were relatively smaller than token vocabulary size. To avoid redundancy XXX, we reduce the embedding dimensions of POSE and PhE to 8 and 32 respectively. The process described in LP is then repeated.

TE_t dimension: 128
POSE_t dimension: 8
PhE_t dimension: 32
Dense = (128 + 8 + 32) input nodes, 128 output nodes

$\text{ATE}_\text{t} = [\text{TE}_\text{t} : \text{POSE}_\text{t} : \text{PhE}_\text{t}]$
$\text{ATEP}_\text{t} = \text{Dense}(\text{ATE}_\text{t})$

QE = BiLSTM(ATEP)

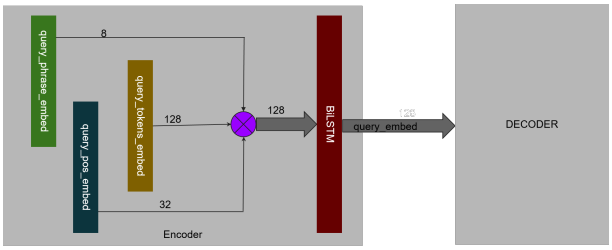

Figure 4: Architecture of Pengcheng's model.

4) Raw Query Independent Preprojection (AdvLP): Rather than applying one linear projection on ATE, we apply two here, where the first is independent of the input query. POSE and PhE are concatenated and projected to create Augmentation Embedding (AE), which is then concatenated with TE and projected to produce QE.

TE_t dimension: 128
$POSE_t$ dimension: 128
PhE_t dimension: 128
Dense = (128 * 2) input nodes, 128 output nodes

AE_t = Dense([$POSE_t$: PhE_t])
ATE_t = [TE_t : AE_t]
$ATEP_t$ = Dense(ATE_t)
QE = BiLSTM(ATEP)

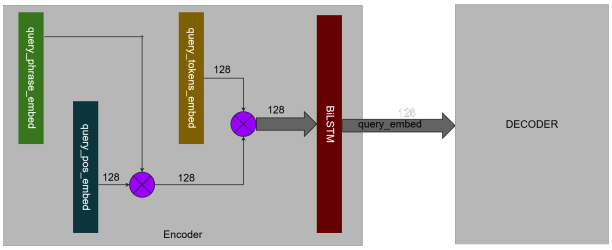

Figure 5: Architecture of Pengcheng's model.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

V. Experiments

All decoder dimensions and configurations are left untouched and are thus the same as in Pengcheng's model. For each of the above models, the embedding sizes are as described Models. Each model is run for a maximum of 50 epochs, with early stopping if the validation metrics do not change for 10 epochs.

VI. Results

Models		Metrics	
		BLUE	Accu.
Pengcheng		84.5	71.6
Base		73.2	67.9
NL2code	BC	73.6	69.4
	LP	<u>74.3</u>	<u>69.7</u>
	LPrd	73.6	69.0
	AdvLP	73.7	69.1

VII. Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.