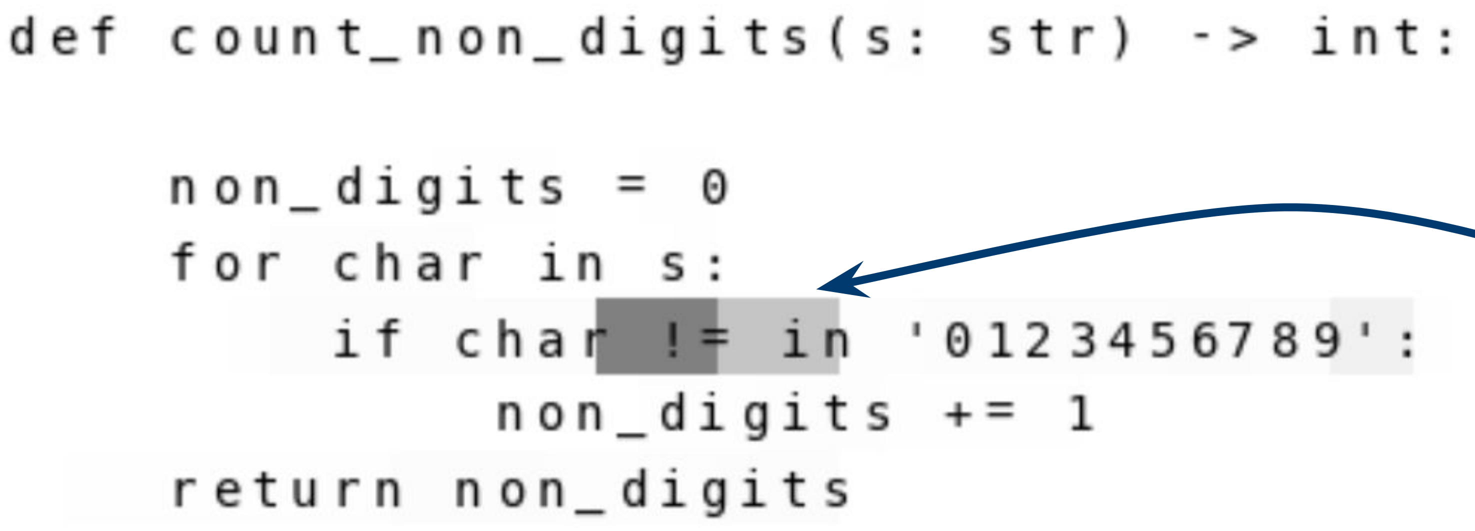
Using Deep Learning to Localize Errors in Student Code Submissions

Shion Fujimori
Mohamed Harmanani
Owais Siddiqui
Lisa Zhang

UNIVERSITY OF TORONTO

Goal

Build Deep Learning models that automatically highlight errors in student submissions to CS1 Python coding problems.



```
def count_non_digits(s: str) -> int:

    non_digits = 0
    for char in s:
        if char != in '0123456789':
            non_digits += 1
    return non_digits
```

Syntactic Issue

Questions

1. Can we build deep learning models that automatically highlight errors in student submissions to CS1 Python coding problems?

2. To what extent can an automated metric effectively measure the models' ability to localize errors?

Data and Context

We use the submissions to Python programming problems from our CS1 course from 2015-2019 to train and test our models.

We focus on 3 of the programming problems

Data Collection

```
def contains_no_lowercase_vowels(phrase):
    for ch in phrase:
        if ch in vowels:
            return False
    return True
```

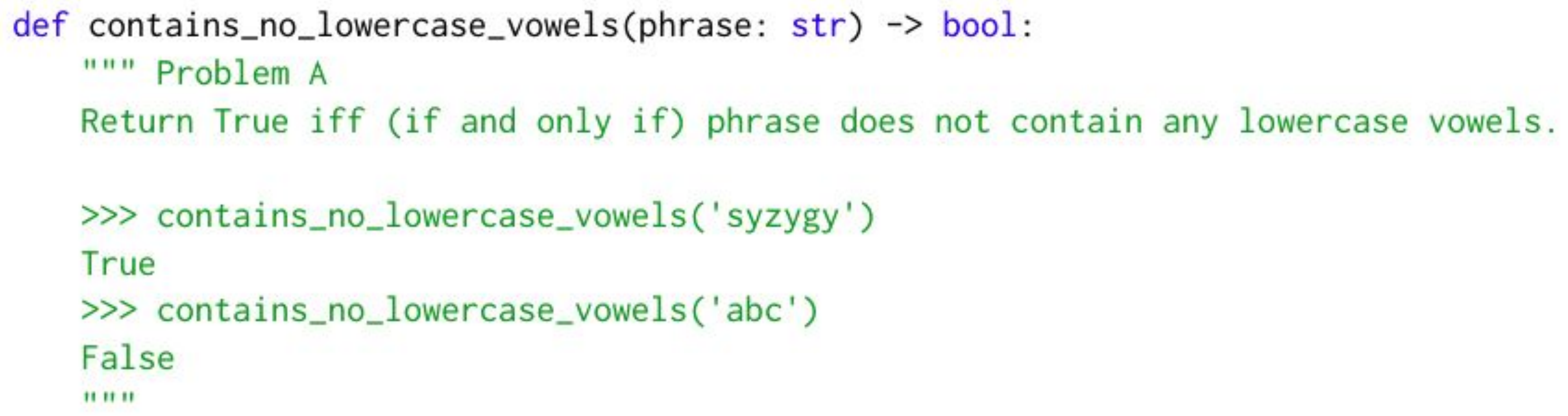
Incorrect submission

Change in the code

```
def contains_no_lowercase_vowels(phrase):
    for ch in phrase:
        if ch in “aeiou”:
            return False
    return True
```

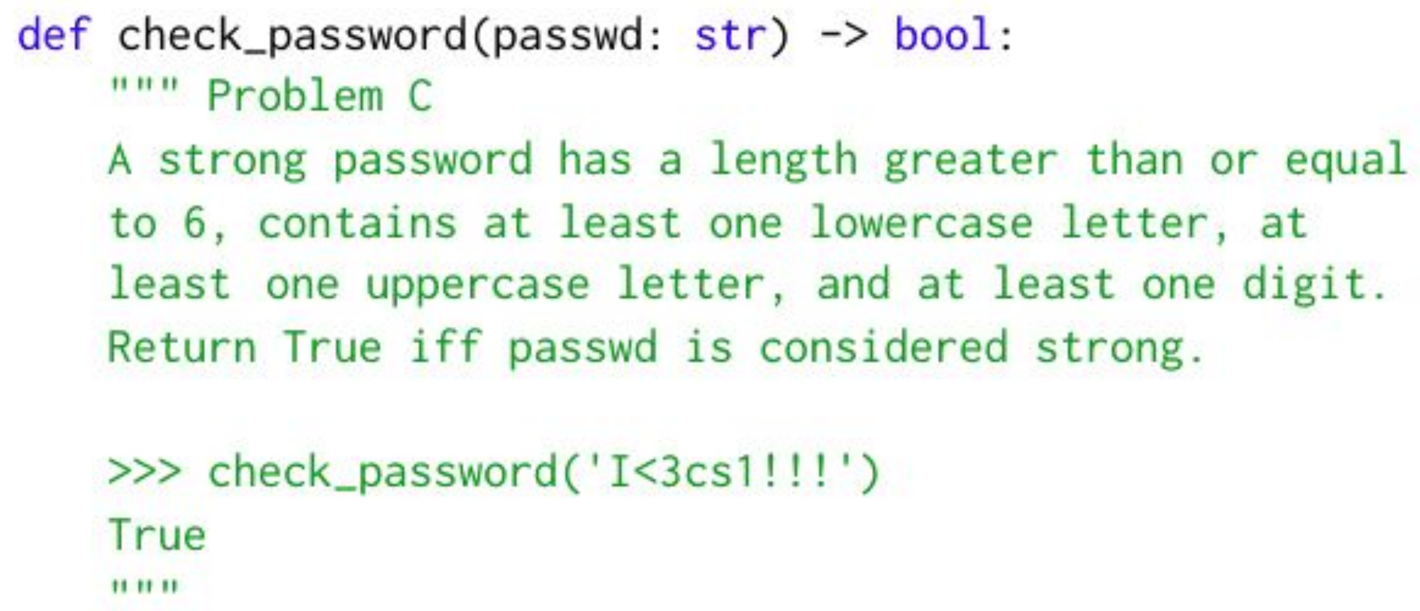
Correct submission

Problems



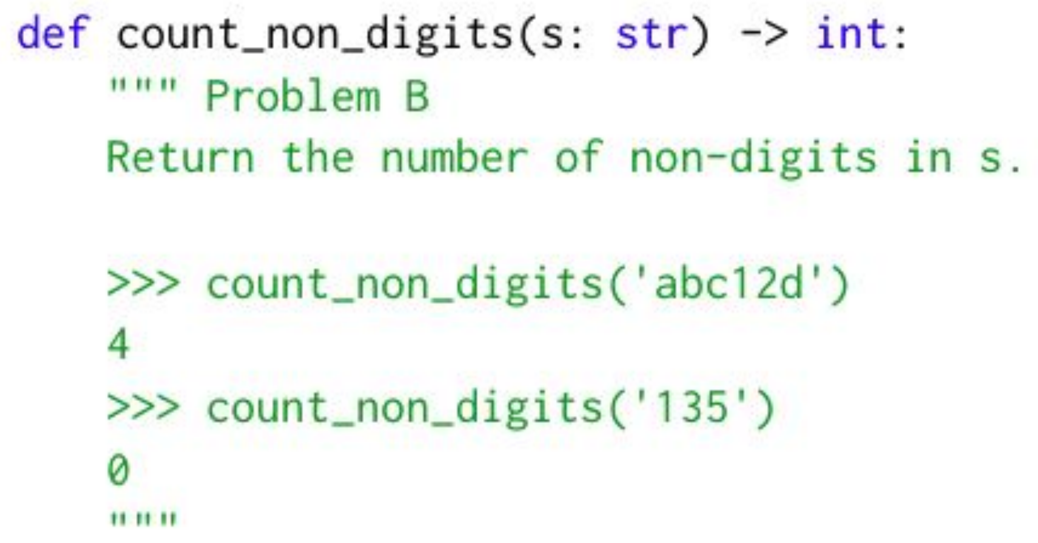
```python
def contains_no_lowercase_vowels(phrase: str) -> bool:
    """ Problem A
    Return True iff (if and only if) phrase does not contain any lowercase vowels.

    >>> contains_no_lowercase_vowels('syzygy')
    True
    >>> contains_no_lowercase_vowels('abc')
    False
    """
```



```python
def check_password(passwd: str) -> bool:
    """ Problem C
    A strong password has a length greater than or equal
    to 6, contains at least one lowercase letter, at
    least one uppercase letter, and at least one digit.
    Return True iff passwd is considered strong.

    >>> check_password('I<3cs1!!!')
    True
    """
```



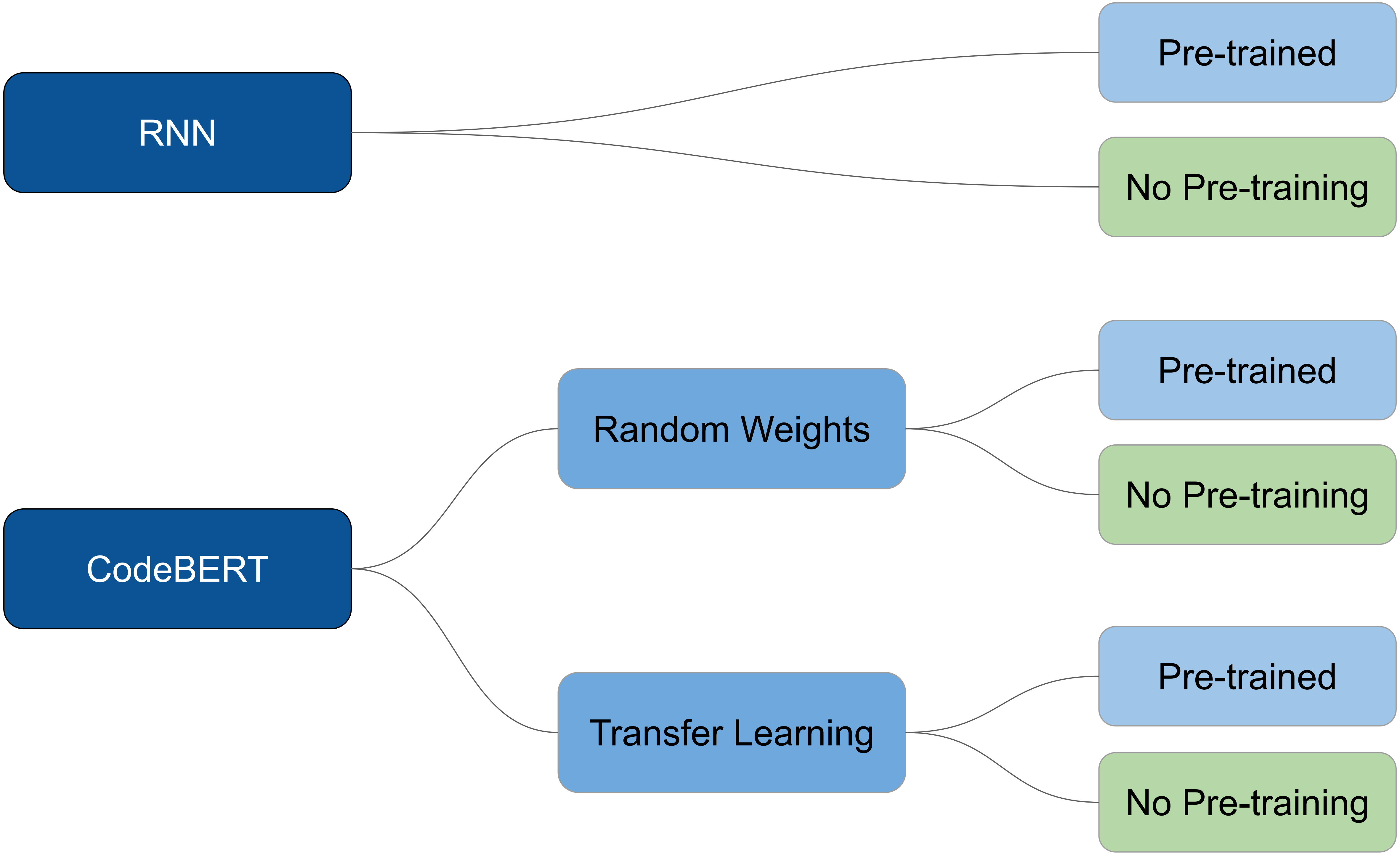
```python
def count_non_digits(s: str) -> int:
    """ Problem B
    Return the number of non-digits in s.

    >>> count_non_digits('abc12d')
    4
    >>> count_non_digits('135')
    0
    """
```

Data Size

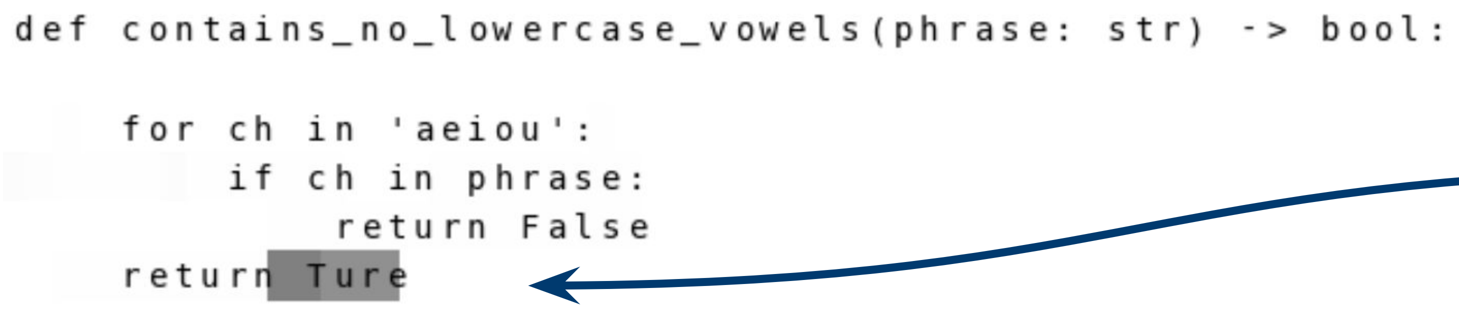
	Number of data pairs			
Problem:	A	B	C	All
Training (2015-2018)	2,593	2,586	2,011	15,006
Testing (2019)	566	656	568	1,793

Deep Learning Models we explore



Evaluation Method

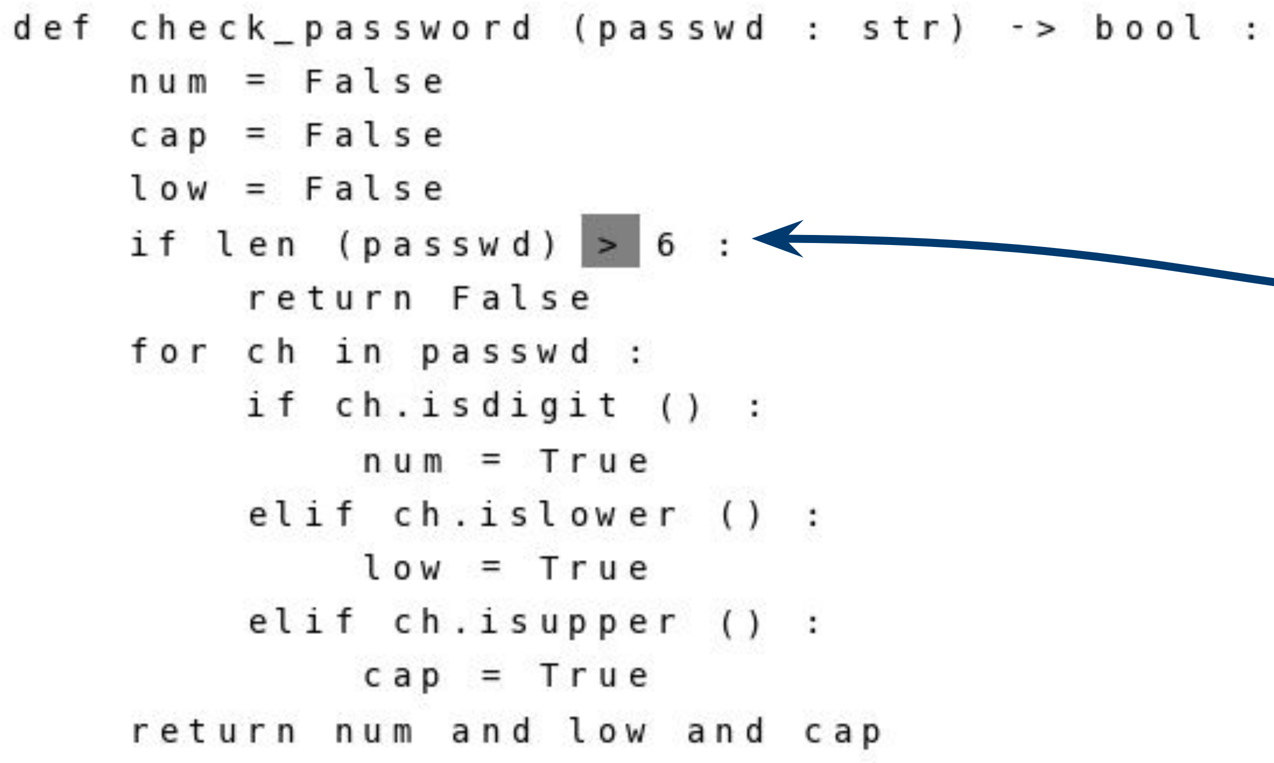
Two types of issues



```
def contains_no_lowercase_vowels(phrase: str) -> bool:

    for ch in 'aeiou':
        if ch in phrase:
            return False
    return Ture
```

Syntactic



```
def check_password (passwd : str) -> bool :
    num = False
    cap = False
    low = False
    if len (passwd) > 6 :
        return False
    for ch in passwd :
        if ch.isdigit () :
            num = True
        elif ch.islower () :
            low = True
        elif ch.isupper () :
            cap = True
    return num and low and cap
```

Semantic

Evaluation Method

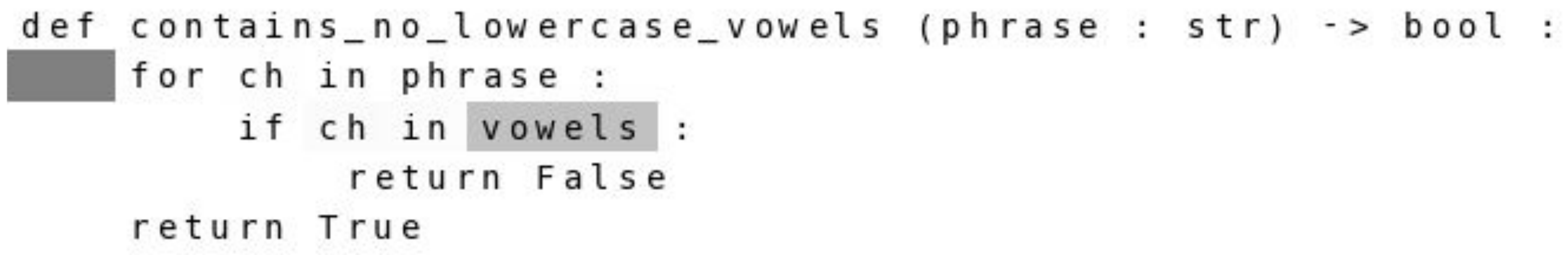
Two evaluation metrics to measure the model performance:

1) **AUC score**
2) **Human evaluation score**

Employ a human evaluation strategy to verify that AUC scores correlate with real-world performance

Human Evaluation Rubric

Behaviour	**Score**
● One correct issue highlighted ● No incorrect issues highlighted	1.0
● The model highlights all or parts of an issue, *AND* ● The model highlights a small number of irrelevant tokens	0.5
● The main issue is not highlighted *OR* ● There are many irrelevant issues highlighted	0.0

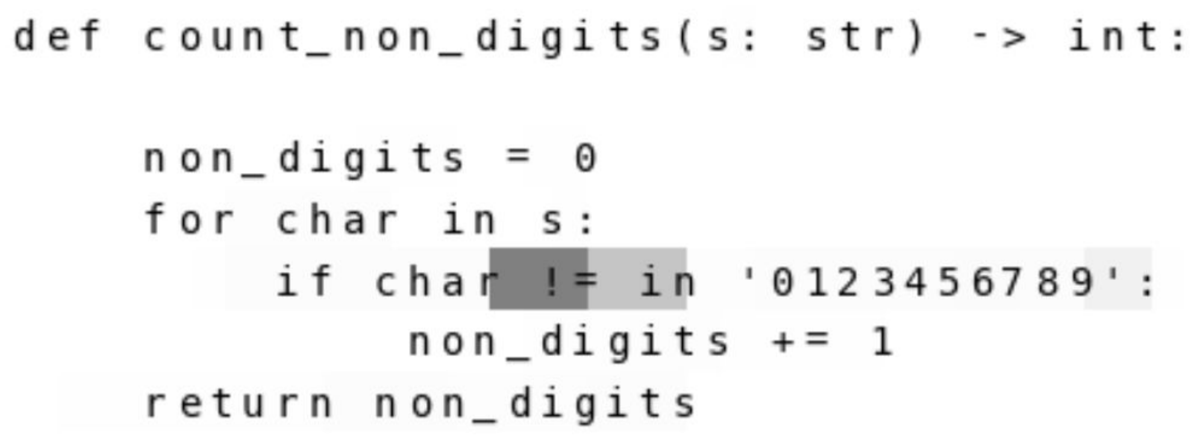


```
def contains_no_lowercase_vowels (phrase : str) -> bool :
    for ch in phrase :
        if ch in vowels :
            return False
    return True
```

RNN output on Problem A

Human Evaluation Rubric

Behaviour	Score
● One correct issue highlighted ● No incorrect issues highlighted	1.0
● The model highlights all or parts of an issue, *AND* ● The model highlights a small number of irrelevant tokens	0.5
● The main issue is not highlighted *OR* ● There are many irrelevant issues highlighted	0.0



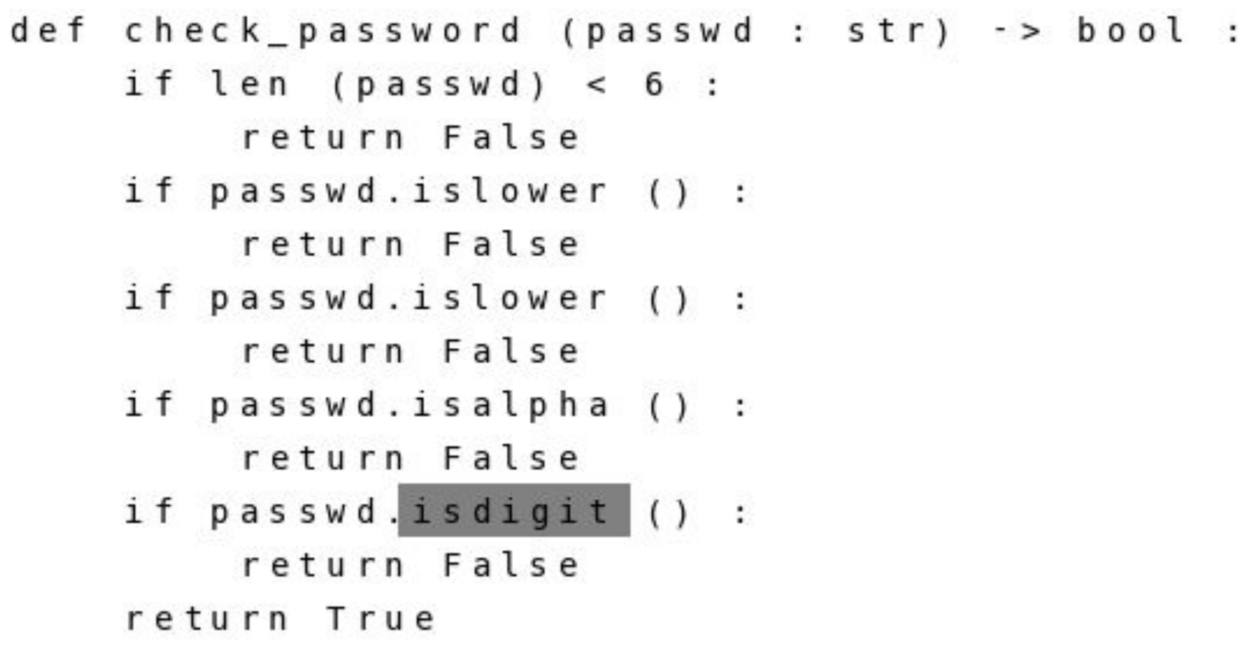
```
def count_non_digits(s: str) -> int:

    non_digits = 0
    for char in s:
        if char != in '0123456789':
            non_digits += 1
    return non_digits
```

CodeBERT output on Problem B

Human Evaluation Rubric

Behaviour	Score
● One correct issue highlighted ● No incorrect issues highlighted	1.0
● The model highlights all or parts of an issue, *AND* ● The model highlights a small number of irrelevant tokens	0.5
● The main issue is not highlighted *OR* ● There are many irrelevant issues highlighted	0.0



```
def check_password (passwd : str) -> bool :
    if len (passwd) < 6 :
        return False
    if passwd.islower () :
        return False
    if passwd.islower () :
        return False
    if passwd.isalpha () :
        return False
    if passwd.isdigit () :
        return False
    return True
```

RNN output on Problem C

Human Evaluation - Syntactic

Model	**Syntactic Test Set**		
Problem:	**A**	**B**	**C**
RNN	0.57	0.66	0.09
RNN-Pretrain	**0.80**	**0.87**	**0.64**
CodeBERT-Transfer	0.55	0.57	0.25
CodeBERT-TransferPretrain	0.69	0.86	0.56

Human Evaluation - Semantic

Model	Semantic Test Set		
Problem:	**A**	**B**	**C**
RNN	0.46	0.61	0.05
RNN-Pretrain	0.68	0.79	0.34
CodeBERT-Transfer	0.58	0.81	0.44
CodeBERT-TransferPretrain	**0.71**	**0.82**	**0.53**

Test AUC - Syntactic

Model	Test AUC			Human Evaluation Score		
Problem:	**A**	**B**	**C**	**A**	**B**	**C**
RNN	0.91	0.90	0.72	0.57	0.66	0.09
RNN-Pretrain	0.91	**0.92**	**0.81**	**0.80**	**0.87**	**0.64**
CodeBERT-Transfer	0.93	0.91	0.77	0.55	0.57	0.25
CodeBERT-TransferPretrain	**0.95**	0.91	0.80	0.69	0.86	0.56

Test AUC - Semantic

Model	Test AUC			Human Evaluation Score		
Problem:	A	B	C	A	B	C
RNN	0.92	0.88	0.75	0.46	0.61	0.05
RNN-Pretrain	**0.93**	0.89	**0.81**	0.68	0.79	0.34
CodeBERT-Transfer	0.92	0.91	0.76	0.58	0.81	0.44
CodeBERT-TransferPretrain	0.92	**0.92**	**0.81**	**0.71**	**0.82**	**0.53**

Conclusion

- Deep Learning models are able to localize errors in student code, with different levels of success
  - Our models performed well on easy problems, but struggled on harder problems
- Automated metrics like AUC may provide limited insights into a model’s real-world performance and behaviour
  - Human evaluation should be taken into account when building Deep Learning tools in a CS Education context

Thank you for listening!

Contact Information

- Shion Fujimori -
University of Toronto
Email: shion.fujimori@mail.utoronto.ca

- Lisa Zhang -
University of Toronto Mississauga
Email: lczhang@cs.toronto.edu