Iby and Aladar Fleischman
Faculty of Engineering
Tel Aviv University

הפקולטה להנדסה
ע"ש איבי ואלדר פליישמן
אוניברסיטת תל-אביב

Guided Data Science

Project Number: 18-1-1-1638

By: Gur Yaniv, Tamir Huber Advisor: Amit Somech

Motivation

Data Science is a difficult job, And the work-flow is similar for many cases.

We want to design a code recommendation system for data scientists.

This system will:

- Use a dataset of existing Data-Science solutions
- Understand the purpose of a given user code (jupyter notebook cell)
- Provide a next-step recommendation (next line of code)

Our Solution

Our solution is consisted of the following **parts**:

1. Dataset Builder

- Automated crawler that downloads notebooks (existing solutions) and all available metadata from Kaggle.com and parses into a csv file
- Built using Selenium and Kaggle API
- Formed a large dataset of Data Science solutions for different problems (could also be useful for other purposes)

2. Workflow Stage Classifier

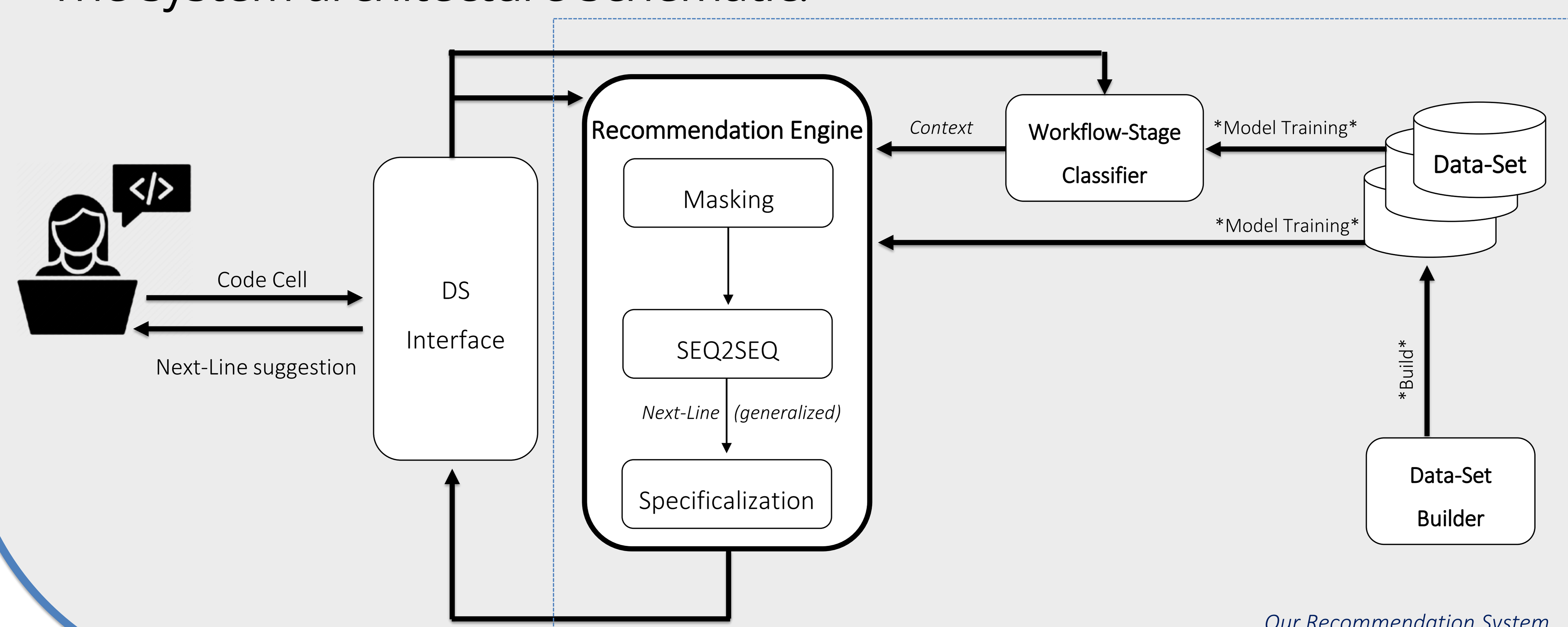
- Tags the data (downloaded notebook code cells) using the new data programming paradigm for weak supervision (Snorkel by Stanford).
- LSTM classifier that given a jupyter notebook code cell, classifies it to the relevant Data Science workflow stage.
- Essentially, we can understand the purpose of a given code.

3. Recommendation Engine

- Sequence-to-Sequence model, using a bidirectional GRU as an encoder and another RNN with a global attention mechanism (ATTN)
- Basically, a Chatbot that gets input from the user (a cell of code) and outputs a recommendation for the next line of code based on our dataset

Recommendation Engine Flow:
Masking, Getting the workflow stage → choosing the relevant S2S model according to the workflow stage → Getting a generalized recommendation from the model → Specification

The system architecture schematic:



Highlights and Example

Loss Function:

- Not all tokens are relevant, we use masked loss.
- Cross-Entropy:

$$H(y,p) = -\sum_i y_i log(p_i)$$

Code Masking:

- Not all code parts are relevant for our purpose
- We mask our Data using the code's AST
- We use a summarized representation of relevant data

Specification:

- We want to tailor the recommendation to get useable code for the user.
- Keep track of variables

Recommendation Example:

```
>> USER:
df=pd.read_csv('../clicks_train.csv')
df.groupby('id')['ad_id'].count().value_counts()
```

*User's code's workflow stage is: Data Exploration

```
>> BOT:
x=df.groupby('id')['ad_id'].count().value_counts()
sns.barplot(x.index, x.values, alpha=0.8)
```

*The system recognizes the context and the user gets an exploratory visualization

⋮