My Kaggle Journey

Angie Sheng

Two Kaggle Competitions So Far...

- **Google QUEST Q&A Labeling - Silver Medal**

Improving automated understanding of complex question answer content

- **M5 Forecasting - Accuracy - Top 11%**

Estimate the unit sales of Walmart retail goods

Google Q&A Labeling - A Glimpse of the Data

	qa_id	question_title	question_body	question_user_name	question_user_page	answer	answer_user_name
0	0	What am I losing when using extension tubes in...	After playing around with macro photography on...	ysap	https://photo.stackexchange.com/users/1024	I just got extension tubes, so here's the skin...	rfusca
1	1	What is the distinction between a city and a s...	I am trying to understand what kinds of places...	russellpierce	https://rpg.stackexchange.com/users/8774	It might be helpful to look into the definitio...	Erik Schmidt

6071 question & answer pairs, asked by 3215 different users and answered by 4114 different users

Google Q&A Labeling - A Glimpse of the Data

answer_user_page	url	category
https://photo.stackexchange.com/users/1917	http://photo.stackexchange.com/questions/9169/...	LIFE_ARTS
https://rpg.stackexchange.com/users/1871		

question_well_written	answer_helpful	answer_level_of_information	answer_plausible
1.000000	1.000000	0.666667	1.000000
0.888889	0.888889	0.555556	0.888889

Ratings based on Q&A pair (30 dimensions)

Google Q&A Labeling - Key insights from EDA

Categorical Features

One-hot Encoding

```
train_df['category'].unique()
```

```
array(['LIFE_ARTS', 'CULTURE', 'SCIENCE', 'STACKOVERFLOW', 'TECHNOLOGY'],
      dtype=object)
```

```
train_df['category'].nunique()
```

```
5
```

```
train_df['host'].nunique()
```

```
63
```

```
train_df['host'].unique()
```

```
array(['photo.stackexchange.com', 'rpg.stackexchange.com',
       'electronics.stackexchange.com', 'judaism.stackexchange.com',
       'graphicdesign.stackexchange.com', 'stackoverflow.com',
       'askubuntu.com', 'gaming.stackexchange.com', 'serverfault.com',
ge.com',
ackexchange.com',
ange.com',
ackexchange.com',
change.com',
stackexchange.com',
com',
xchange.com',
ange.com',
xchange.com',
       'raspberrypi.stackexchange.com', 'academia.stackexchange.com',
       'bicycles.stackexchange.com', 'android.stackexchange.com',
       'mathoverflow.net', 'boardgames.stackexchange.com',
       'movies.stackexchange.com', 'anime.stackexchange.com',
       'apple.stackexchange.com', 'webmasters.stackexchange.com',
       'diy.stackexchange.com', 'gis.stackexchange.com',
       'stats.stackexchange.com', 'ux.stackexchange.com',
       'english.stackexchange.com', 'scifi.stackexchange.com',
       'gamedev.stackexchange.com', 'cs.stackexchange.com',
```

Google Q&A Labeling - Key insights from EDA

Targets could be put into two groups, which means we could train 2 models for each group

```
train_df.columns[11:32] #targets that are related to question and question body

Index(['question_asker_intent_understanding', 'question_body_critical',
       'question_conversational', 'question_expect_short_answer',
       'question_fact_seeking', 'question_has_commonly_accepted_answer',
       'question_interestingness_others', 'question_interestingness_self',
       'question_multi_intent', 'question_not_really_a_question',
       'question_opinion_seeking', 'question_type_choice',
       'question_type_compare', 'question_type_consequence',
       'question_type_definition', 'question_type_entity',
       'question_type_instructions', 'question_type_procedure',
       'question_type_reason_explanation', 'question_type_spelli
       'question_well_written'],
      dtype='object')
```

21 for title-body pair model

```
train_df.columns[32:] #targets that are related to answers

Index(['answer_helpful', 'answer_level_of_information', 'answer_plausible',
       'answer_relevance', 'answer_satisfaction', 'answer_type_instructions',
       'answer_type_procedure', 'answer_type_reason_explanation',
       'answer_well_written'],
      dtype='object')
```

9 for title-answer pair model

Google Q&A Labeling - Key insights from EDA

Range: [0,1]
Ratings are discrete

Multi-label Classification & Post-Processing

```
train_df.iloc[:,11:].nunique()
```

```
question_asker_intent_understanding      9
question_body_critical                   9
question_conversational                  5
_expect_short_answer                     5
_fact_seeking                            5
_has_commonly_accepted_answer            5
_interestingness_others                  9
_interestingness_self                    9
_multi_intent                            5
_not_really_a_question                   5
_opinion_seeking                         5
_type_choice                             5
_type_compare                            5
_type_consequence                        5
_type_definition                         5
_type_entity                             5
_type_instructions                       5
_type_procedure                          5
_type_reason_explanation                 5
_type_spelling                           3
_well_written                            9
```

```
train_df.iloc[:,11:].max()
```

```
question_asker_intent_understanding      1.000000
question_body_critical                   1.000000
question_conversational                  1.000000
ver                                      1.000000
                                         1.000000
pted_answer                              1.000000
thers                                    1.000000
elf                                      1.000000
                                         1.000000
tion                                     1.000000
                                         1.000000
                                         1.000000
                                         1.000000
                                         1.000000
                                         1.000000
                                         1.000000
                                         1.000000
```

```
train_df.iloc[:,11:].min()
```

```
question_asker_intent_understanding      0.333333
question_body_critical                   0.333333
question_conversational                  0.000000
question_expect_short_answer             0.000000
question_fact_seeking                    0.000000
question_has_commonly_accepted_answer    0.000000
question_interestingness_others          0.333333
question_interestingness_self            0.333333
question_multi_intent                    0.000000
question_not_really_a_question           0.000000
```

Google Q&A Labeling - Key insights from EDA

```
train_df['question_title_length'] = train_df['question_title'].str.split(' ').map(lambda x: len(x))
train_df['question_body_length'] = train_df['question_body'].str.split(' ').map(lambda x: len(x))
train_df['answer_length'] = train_df['answer'].str.split(' ').map(lambda x:len(x))
```

```
train_df['total_length'] = train_df['question_title_length'] + train_df['answer_length']
```

```
train_df['total_length'].describe()
```

```
count     6079.000000
mean       152.744695
std        206.109610
min          7.000000
25%         57.000000
50%        100.000000
75%        180.000000
max       8177.000000
Name: total_length, dtype: float64
```

Max input length for BERT & RoBERTa -> 512

Need to trim!

Google Q&A Labeling - Feature Engineering

1. **One-hot encoding for url & category**

```
find = re.compile(r"^[^.]*")

train['netloc'] = train['url'].apply(lambda x: re.findall(fi
```

```
train[['netloc','url']]
```

	netloc	url
0	photo	http://photo.stackexchange.com/questions/9169/...
1	rpg	http://rpg.stackexchange.com/questions/47820/w...
2	electronics	http://electronics.stackexchange.com/questions...

```
train_df['category'].unique()
```

```
array(['LIFE_ARTS', 'CULTURE', 'SCIENCE', 'STACKOVERFLOW', 'TECHNOLOGY'],
      dtype=object)
```

```
train_df['category'].nunique()
```

```
5
```

```
train['netloc'].unique()
```

```
array(['photo', 'rpg', 'electronics', 'judaism', 'graphicdesign',
       'stackoverflow', 'askubuntu', 'gaming', 'serverfault', 'uni
       'dba', 'codereview', 'crypto', 'tex', 'travel', 'webapps',
       'mechanics', 'physics', 'math', 'programmers', 'biology',
       'wordpress', 'superuser', 'music', 'blender', 'dsp', 'drupa
       'meta', 'security', 'raspberrypi', 'academia', 'bicycles',
       'android', 'mathoverflow', 'boardgames', 'movies', 'anime',
       'apple', 'webmasters', 'diy', 'gis', 'stats', 'ux', 'englis
       'scifi', 'gamedev', 'cs', 'cooking', 'sharepoint', 'mathema
       'salesforce', 'expressionengine', 'magento', 'christianity'
       'chemistry', 'money', 'ell', 'robotics', 'softwarerecs'],
      dtype=object)
```

```
train['netloc'].nunique()
```

```
59
```

Google Q&A Labeling - Feature Engineering

2. Universal Sentence Encoder -> Cosine Similarities & L2 Distance

The Universal Sentence Encoder (USE) encodes text into high dimensional vectors that can be used for diverse tasks. The input is the variable-length English text, and the output is a 512-dimensional vector.

Question Title **Question Body**

Universal Sentence Encoder

512-dimensional vector 512-dimensional vector

Cosine Similarities & L2 Distance

Question Title **Answer**

Universal Sentence Encoder

512-dimensional vector 512-dimensional vector

Cosine Similarities & L2 Distance

Google Q&A Labeling - Feature Engineering

3. Summary of Data Engineering

a. One-hot encoding of URL & Category (64)

b. Universal Sentence Encoder (512+512)

c. Cosine Similarities (1)

d. L2 Distance (1)

1090 new features in total

Google Q&A Labeling - Pre-Processing

```
def _trim_input(title, question, max_sequence_length,
                t_max_len=100, head=128, tail=281, Q=True, q_max_len=239, a_max_len=239):
    t = tokenizer.tokenize(title)
    q = tokenizer.tokenize(question)
    # a = tokenizer.tokenize(answer)

    t_len = len(t)
    q_len = len(q)
    # a_len = len(a)

    if (t_len + q_len + 4) > max_sequence_length:
        if t_max_len > t_len:
            t_new_len = t_len
            q_head = head
            q_tail = 508 - q_head - t_new_len
        else:
            t_new_len = t_max_len
            if (t_new_len + q_len + 4) > max_sequence_length:
                q_head = head
                q_tail = 508 - q_head - t_new_len
```

Trimming:

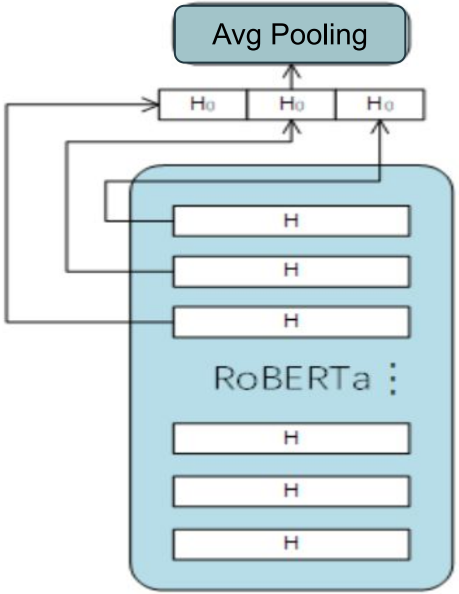
Q Title + Q Body/Answer < 508

Strategy: Head + Tail

```
token = ["<s>"] + title + ["</s>"] + ["</s>"] + question + ["</s>"]
```

Google Q&A Labeling - Model Structure

batch_size x 512 x 768
(*batch _size x maxseqlen x emb_size*)



1090 New Features

Fully Connected Layer

21 units for title-body pair model

9 units for title-answer pair model

Google Q&A Labeling - Training & Prediction

Customized Learning Rate: A customized scheduler inherited from PolynomialDecay was used here to change the learning rate dynamically.

GroupKFold: Did 8-fold cv for 8 epochs and saved the training weights with the highest cv scores.

Optimizer: Adam optimizer with mixed-precision data types, which dynamically and automatically adjusting the scaling to prevent Inf or NaN values and saved training time.

Post-Processing: Discretization based on different target (For Competition Purpose!)

The Final Predictions are the average of 8 pairs of Roberta models (8-fold).

M5 Forecasting

The goal:

- To **predict sales data** provided by the retail giant Walmart **28 days** into the future.

The data:

1. We are working with **30,490 hierarchical time series.** The data were obtained in the 3 US states of California (CA), Texas (TX), and Wisconsin (WI). The sales information reaches back from Jan 2011 to June 2016.
2. In addition to the sales numbers, we are also given corresponding data on prices, promotions, and holidays.

M5 Forecasting - A Glimpse of Data

id	item_id	dept_id	cat_id	store_id	state_ic
HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA
HOBBIES_1_002_CA_1_validation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA
HOBBIES_1_003_					
HOBBIES_1_004_					
HOBBIES_1_005_					

d_1905	d_1906	d_1907	d_1908	d_1909	d_1910	d_1911	d_1912	d_1913
3	0	1	1	1	3	0	1	1
0	0	0	0	1	0	0	0	0
1	2	1	1	1	0	1	1	1
								2
								4

sales_train.csv: The data comprises 3049 individual products from *3 categories* and *7 departments*, sold in *10 stores* in *3 states*. It has 1 column for each of the 1941 days from 2011-01-29 and 2016-05-22. The number of rows is 30490 for all combinations of 3049 items and 10 stores.

Walmart

M5 Forecasting - A Glimpse of Data

store_id	item_id	wm_yr_wk	sell_price
CA_1	HOBBIES_1_001	11325	9.58
CA_1	HOBBIES_1_001	11326	9.58
CA_1	HOBBIES_1_001	11327	8.26
CA_1	HOBBIES_1_001	11328	8.26
CA_1			

sell_prices.csv : Provides the store and item IDs together with the sales price of the item as a weekly average.

Walmart

M5 Forecasting - A Glimpse of Data

date	wm_yr_wk	weekday	wday	month	year	d	event_name_1	event_type_1	event_name_2	event_type_2	snap_CA	snap_TX	snap_WI
2011-01-29	11101	Saturday	1	1	2011	d_1	NA	NA	NA	NA	0	0	0
2011-01-30	11101	Sunday	2	1	2011	d_2	NA	NA	NA	NA	0	0	0
2011-01-31	11101	Monday	3	1	2011	d_3	NA	NA	NA	NA	0	0	0
2011-02-01	11101	Tuesday	4	2	2011	d_4	NA	NA	NA	NA	1	1	0
2011-02-02	11												1

calendar.csv: The calendar data gives us date features such as weekday, month, or year; alongside 2 different event features and a SNAP* food stamps flag.

*SNAP: federal nutrition assistance program for low-income individuals and families)

M5 Forecasting - Feature Engineering

Lag and sliding window features:

```python
def create_fea(dt):
    lags = [7, 28]
    lag_cols = [f"lag_{lag}" for lag in lags ]
    for lag, lag_col in zip(lags, lag_cols):
        dt[lag_col] = dt[["id","sales"]].groupby("id")["sales"].shift(lag)

    wins = [7, 28]
    for win in wins :
        for lag,lag_col in zip(lags, lag_cols):
            dt[f"rmean_{lag}_{win}"] = dt[["id", lag_col]].groupby("id")[lag_col].transform(lambda x : x.rolling(win).mean())
```

M5 Forecasting - Feature Engineering

[101]:
```
dt.loc[dt.id == "HOBBIES_1_001_CA_1_validation",
       ["sales", "lag_7", "lag_28", "rmean_7_7", "rmean_7_28", "rmean_28_7", "rmean_28_28"]
      ].head(56).reset_index(drop=True).style.applymap(lambda x: "background-color: yellow" if np.isnan(x) else "")
```

[101]:

	sales	lag_7	lag_28	rmean_7_7	rmean_7_28	rmean_28_7	rmean_28_28
0	0.000000	nan	nan	nan	nan	nan	nan
1	0.000000	nan	nan	nan	nan	nan	nan
2	0.000000	nan	nan	nan	nan	nan	nan
3	0.000000	nan	nan	nan	nan	nan	nan
4	0.000000	nan	nan	nan	nan	nan	nan
5	1.000000	nan	nan	nan	nan	nan	nan
6	1.000000	nan	nan	nan	nan	nan	nan
7	0.000000	0.000000	nan	nan	nan	nan	nan
8	2.000000	0.000000	nan	nan	nan	nan	nan
9	0.000000	0.000000	nan	nan	nan	nan	nan
10	0.000000	0.000000	nan	nan	nan	nan	nan
11	1.000000	0.000000	nan	nan	nan	nan	nan
12	0.000000	1.000000	nan	nan	nan	nan	nan
13	0.000000	1.000000	nan	0.285714	nan	nan	nan
14	0.000000	0.000000	nan	0.285714	nan	nan	nan
15	1.000000	2.000000	nan	0.571429	nan	nan	nan
16	0.000000	0.000000	nan	0.571429	nan	nan	nan
17	1.000000	0.000000	nan	0.571429	nan	nan	nan
18	0.000000	1.000000	nan	0.714286	nan	nan	nan
19	0.000000	0.000000	nan	0.571429	nan	nan	nan
20	0.000000	0.000000	nan	0.428571	nan	nan	nan
21	0.000000	0.000000	nan	0.428571	nan	nan	nan
22	1.000000	1.000000	nan	0.285714	nan	nan	nan
23	0.000000	0.000000	nan	0.285714	nan	nan	nan
24	1.000000	1.000000	nan	0.428571	nan	nan	nan
25	0.000000	0.000000	nan	0.285714	nan	nan	nan
26	1.000000	0.000000	nan	0.285714	nan	nan	nan
27	0.000000	0.000000	nan	0.285714	nan	nan	nan
28	0.000000	0.000000	0.000000	0.285714	nan	nan	nan
29	1.000000	1.000000	0.000000	0.285714	nan	nan	nan
30	0.000000	0.000000	0.000000	0.285714	nan	nan	nan
31	0.000000	1.000000	0.000000	0.285714	nan	nan	nan
32	1.000000	0.000000	0.000000	0.285714	nan	nan	nan
33	1.000000	1.000000	1.000000	0.428571	nan	nan	nan
34	2.000000	0.000000	1.000000	0.428571	0.357143	0.285714	nan
35	0.000000	0.000000	0.000000	0.428571	0.357143	0.285714	nan
36	1.000000	1.000000	2.000000	0.428571	0.392857	0.571429	nan
37	0.000000	0.000000	0.000000	0.428571	0.392857	0.571429	nan
38	0.000000	0.000000	0.000000	0.285714	0.392857	0.571429	nan
39	0.000000	1.000000	1.000000	0.428571	0.428571	0.714286	nan

*For simplicity a month is 28 days

- How the sales were last friday compared to this friday?
- How the sales were first weekend of the last month compared to first weekend of this month?
- Comparing last saturday to this saturday is too specific. We want to capture the whole week/month and not just a single day sale comparison, bringing the `lag_7 or lag_28` value into "better weekly/monthly context"

	sales	lag_7	lag_28	rmean_7_7	rmean_7_28	rmean_28_7	rmean_28_28
D1	0.000000	nan	nan	nan	nan	nan	nan
D14	0.000000	1.000000	nan	0.285714	nan	nan	nan
D35	2.000000	0.000000	1.000000	0.428571	0.357143	0.285714	nan
D56	0.000000	2.000000	0.000000	1.142857	0.607143	0.428571	0.357143

- rmean_7_7: the sales of the whole *previous week ending 7 days in the past*
- *rmean_7_28:* the sales of the entire *previous 4 weeks ending 7 days in the past*
- rmean_28_7: the sales of the whole previous *week ending 4 weeks in the past*
- *rmean_28_28:* the sales of the entire *previous 4 weeks ending 4 weeks in the past*

M5 Forecasting - Data for Training

	id	item_id	dept_id	store_id	cat_id	state_id
0	HOBBIES_1_002_CA_1_validation	1	0	0	0	0
1	HOBBIES_1_0					
2	HOBBIES_1_0					
3	HOBBIES_1_0					
4	HOBBIES_1_0					

event_name_1	event_type_1	event_name_2	event_type_2	snap_CA	snap_TX	snap_WI
0	0	0	0	0.0	1.0	0.0
0	0	0	0	0.0	1.0	0.0
0	0	0	0	0.0	1.0	0.0
0	0	0	0	0.0	1.0	0.0
0	0	0	0	0.0	1.0	0.0

d	sales	date	wm_yr_wk	weekday	wday	month	year	week	quarter	mday
d_350	0.0	2012-01-13	11150	0	7	1	2012	2	1	13
d_350	2.0	2012-01-13	11150	0	7	1	2012	2	1	13
d_350	0.0									
d_350	0.0									
d_350	2.0									

sell_price	lag_7	lag_28	rmean_7_7	rmean_28_7	rmean_7_28	rmean_28_28
3.97	NaN	NaN	NaN	NaN	NaN	NaN
4.34	NaN	NaN	NaN	NaN	NaN	NaN
2.48	NaN	NaN	NaN	NaN	NaN	NaN
0.50	NaN	NaN	NaN	NaN	NaN	NaN
1.77	NaN	NaN	NaN	NaN	NaN	NaN

M5 Forecasting - Model

- **Model: A single LightGBM with Tweedie as Loss Function**
- **Recursive Multi-step Forecast**

```
prediction(t+1) = model(obs(t-1), obs(t-2), ..., obs(t-n))
prediction(t+2) = model(prediction(t+1), obs(t-1), ..., obs(t-n))
```