Predict the relevance of search results using traditional and advance techniques

Hussnain Khalid (huskh803)

Abstract

This research is about the performance of search engine algorithms that use different NLP methods in order to give more relevant results for a specific query. Different traditional machine learning models and some neural network models are used to measure the relevance of search results. In this research, such models were applied to query based on product title and product description to predict median relevance. Performance analysis is based on accuracy and model complexity.

1 Introduction

Today ecommerce business is fastest growing business, and the amount of data is growing exponentially, it becomes more and more important to be able to find relevant information. Search engines are designed to fetch information relevant to search queries. A query is used based on algorithms to fetch optimized search results. Search engine performance is most important for ecommerce business. In order to be efficient different techniques are used to fetch desired data. But it's important to focus on the quality of data instead of quantity of the data. Natural language processing (NLP) is used to research on how to make computers understand human language. As the use of search engines and the amount of processing data increases it becomes very important to handle search queries. Today search engines uses NLP for lemmatization of words in a query to fetch related result to original keywords. So, it's important to measure the relevance of search results to make the search engine more powerful and efficient. One of the important task for this research is also to find out how advanced methods like neural networks can help to predict the relevance of search results.

2 Theory

2.1 Multinomial Naive Bayes

Naive Bayes Classifier is a probabilistic algorithm based on Bayes' theorem with the "naive" assumption of conditional independence between every pair of a feature. This model was used frequently during the course but the reason for mentioning and using this model is that it performed much better than other models.

2.2 Support Vector Machine (SVM)

SVM is another basic model that determines the best decision boundary between vectors that belong to a given group and vectors that do not belong to it. SVM is used because it proves to be very effective model[1]. It works almost same as Naive Bayes Classifier but still few differences which will be explained later.

2.3 Random forest

Random forest module is implemented, as a bagging method, it generates number of trees instead of single tree. The training procedure is shown below.

1. The number of trees(estimators) in the forest (10,50,100).
2. The maximum depth of the tree (4).
3. The minimum number of samples required to split (10).

Grid search is used to find the best estimators but still results are not better than other traditional methods. It became useless as our data is quite biased.

[1] https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-baye

2.4 Multi-layer Perceptron regressor (MLP)

MLP[2] is a neural network that has 3 or more layers of perceptrons (input layer, hidden layer and output layer)[3]. It's a single direction model.

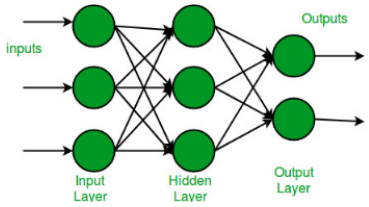


Figure 1: MLP topology

2.5 Elastic-Net

ET[4] is an ensemble approach similar to Random Forest but there are few differences, like Random Forest uses bagging technique to select variation of training data on the other hand ET uses entire dataset for training. Extra Trees are much faster as compared to Random Forest with respect to computational cost[5].

2.6 Extra-Trees Regressor (ET)

Elastic-Net is a linear regression model which uses the penalty factor from Lasso and Ridge regression for regularization[6]. The benefit of using Elastic-Net is it keeps both penalties balanced, for better performance.

2.7 BiDirectional Long Short-Term Memory(BiLSTM)

Bidirectional Long Short-Term Memory (BiLSTM) (Hameed and Garcia-Zapirain, 2020) is a variant of the Long Short-Term Memory (LSTM) neural network architecture that is commonly used in natural language processing (NLP) and speech recognition tasks. In traditional LSTM process sequences only in forward direction, whereas in BiLSTM process sequences in both forward and backward directions simultaneously, allowing it to capture context information from past and future inputs. BiLSTM consists of two LSTM layers that operate in opposite directions. The output of each layer is concatenated to form the final output sequence. BiLSTM captures dependencies in input sequence in both directions, making it useful for tasks that require understanding of context. BiLSTM outperform traditional LSTM and other sequence modeling techniques in NLP tasks, and becomes a popular choice for many researchers and practitioners in the field. The concept behind LSTM model is to regulate the cell states by using three gates input, forget and output gates, as shown in the figure 2.

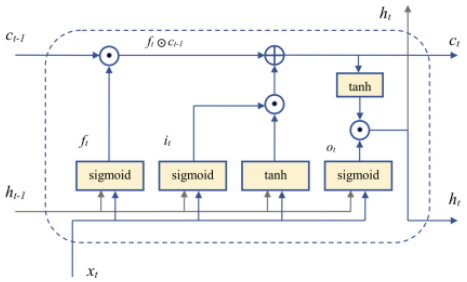


Figure 2: LSTM architecture

The forget gate, f_t, determines whether to forget or to keep information of the previous state, c_{t1} depending on input value, x_t, and the hidden state, h_{t1}, and the output value 0 or 1. The input gate decides how much information of input text, x_t and h_{t1} should pass to update the cell state, and its output to 0 or 1. The value of c_t represents the generated cell state as result of the operations on c_{t1}, f_t and i_t. The output gate, o_t, controls flow of information from the current cell state to hidden state, and its value to 0 or 1, shown in equations below:

$$f_t = sigmoid(W_{fx}x_t + W_{fh}h_{t1} + b_f)$$

$$\begin{aligned} \mathrm{i}_t &= sigmoid(W_{ix}x_t + W_{ih}h_{t1} + b_i) \\ \mathrm{c}_t &= c_{t1}f_t + i_t tanh(W_{cx}x_t + W_{ch}h_{t1} + b_c) \\ \mathrm{o}_t &= sigmoid(W_{ox}x_t + W_{oh}h_{t1} + b_o) \\ \mathrm{h}_t &= o_t tanh(c_t) \end{aligned} \quad (1)$$

Where $x_t \epsilon Rn$ is the input vector, $W \epsilon R^n$, $b \epsilon R^v$ and, n and v depict dimensions of input vector and number of words in the dataset or vocabulary. At any time, t, the inputs to LSTM are input vector x_t, previously hidden state $h_t 1$, and previous cell state c_{t1}, whereas the output is current hidden state h_t and current cell state c_t.

[2] https://www.geeksforgeeks.org/difference-between-multilayer-perceptron-and-linear-regression/

[3] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

[4] https://orbi.uliege.be/bitstream/2268/9357/1/geurts-mlj-advance.pdf

[5] https://towardsdatascience.com/what-when-how-extratrees-classifier-c939f905851c

[6] https://machinelearningmastery.com/elastic-net-regression-in-python/

3 Data

3.1 Raw Data

Data is taken from a Kaggle competition[7]. The size of the data is 10K rows and 5 columns showed below

1. Id
2. Query - Query text.
3. Product title - Product title text
4. Product description - Product description text.
5. Median relevance - Median relevance score.

The Median relevance column contains 4 different classes [4,3,2,1], where 4 is for most relevant product according to the query and 1 is for least relevant product.

3.2 Data Cleaning

Data contains 2444 null values in the product description column. As missing values are only in the product description column we will replace them with empty string (the purpose is explained in the Method section). Data is split into training and test data randomly, where training data is 75 percent and test data is remaining 25 percent. Id column is dropped in most cases as its useless for us mostly.

3.3 Data Example

Training data:

train.head()

	id	query	product_title	product_description	median_relevance
0	0	zippo hand warmer	Zippo Black Polish Hand Warmer, with Filling C...		4
1	1	wall mirrors	Cooper Classics Benedetta Wall Mirrors - 12 di...	The small round shape and shiny nickel finish ...	4
2	2	rachel ray cookware	Rachael Ray Porcelain II Green 10-piece Cookwa...	This versatile 10-piece cookware set from Rach...	4
3	3	flea and tick control for dogs	Hartz UltraGuard Plus Flea and Tick Home Fogge...	details\nThe Hartz UltraGuard Plus Home Fogger...	3
4	4	batman	DC Comics Arkham Asylum Batman Series The Joke...	\n\t \n\t\tEnglish \n\t\t\t\t \n...	2

Figure 3: Training Data

Test data:

test.head()

	id	query	product_title	product_description	median_relevance
0	0	pittsburgh pirates	Fanmats Machine-Made Pittsburgh Pirates Black ...	Show your team pride and add style to your tai...	4
1	1	dish towels	Oasis Cotton Flour Sack Towel (Set of 3)	These Flour Sack Towels\offers the ultimate in ...	3
2	2	portable hard drive	Toshiba Canvio Slim 1 TB External Hard Drive	Toshiba Canvio® Slim II for Mac® Portable Exte...	3
3	3	playstation vita system	Legends Of Chima: Laval's Journey (PlayStation...	Get ready for an epic adventure as Laval races...	2
4	4	galaxy note 3	3M Natural View Screen Protection Film for Sam...	Helps keep your device screen in pristine, lik...	2

Figure 4: Test Data

[7] https://www.kaggle.com/competitions/crowdflower-search-relevance/overview

4 Method

So, in order to evaluate the results, I have to choose a baseline and, in this case, I have to beat 60 percent accuracy. I choose 60 because in the data the relevance of search query with perfect match is around 60 percent (median relevance == 4).

Figure 5: Target Value

We can also call it a dummy classifier which is usually built using the most frequent scheme. Before using any method first, I implement a few techniques to set up training and test data for the model. To do so I dropped the Id column for traditional machine learning models and also combined product title and product description columns together as 1 column so we don't have to drop null rows because of null values present in the product description column. Also, this can help to evaluate the vocabulary better. For preprocessing different techniques are used to remove stop words and some specific stop words added in the list for removal as they are useless for the model training. Also the concept of stemming is used to deal with morphological variants of words. TfidfVectorizer is used to transform the test to feature vectors that is used as an input to our estimators. Pipeline from Sklearn is also used to connect all steps together for different parameters. After all that new training and test data frames are set for implementation of models. And the first traditional and the basic model for this task I used is Naive Bayes which is the most common model and gives an expected good result too. SVM proves to be also a nice classifier in our case there are few differences which will be explained in the result section. I used different penalty parameters of error term but C=10 proves to be best of all these. There is no way to know the best value for C. It only depends upon what result you will have after applying on test data. Random Forest Classifier is not up to the mark and performed really badly. Though different parameters were applied like max depth = 4, min samples leaf = 10 and different estimators

for number of trees (10, 50, 100) are used with a grid search approach for the best estimator. Other than these traditional methods some advanced complex methods are also applied to check how they react but all of them performed really badly. In the discussion section I'll explain what could be the reason for that according to my experiments. Multi-layer Perceptron classifier is used with different setups shown below, all of these techniques are applied separately on validation data.

Hidden layers	Random state	Max iter	Activation	Learning rate	Learning rate init	Early stopping
1	1	500	-	-	-	-
9	1	500	relu	-	-	-
9	1	500	identity	-	-	-
9	1	500	tanh	-	-	-
9	1	500	tanh	adaptive	0.0001	True

Figure 6: MLP Training Summary

In the last configurations activation function 'tanh' proves to be best for this task, 'adaptive' learning rate was used to keep the learning rate constant as long as training loss keeps decreasing and early stopping is also used to terminate training when validation score is not improving[8]. Extra-Trees regressor with following configuration is used n estimators = 250 and min samples split = 10 means number of trees in the forest are 250 and minimum number of samples required to split is 10. Elastic Net[9] is used with these configurations alpha = 0.01, l1 ratio = 0.5 and fit intercept = False. Where alpha is the constant multiplies the penalty term. With l1 ration = 0.5 we can use both L1 and L2 penalty combination and fit intercept is set to be false as it is used to check if the data is already centered or not. All these advanced methods MLP, Extra-Trees and Elastic Net are used as ensemble learning[10], where their predictions are combined together to improve the results.

For BiLSTM three tests are performed on validation data instead of hyperparameters tuning and then the result is compared with the by evaluating the test data. Three separate tests are performed without Gridsearch approach as it was taking a lot of memory and due to insufficient space and processing power tests were taken separately. Below mentioned are the tests run performed using BiLSTM approach.

1. LSTM units = 32, dropout = 0.4, embedding dim = 300 and L2 regularizer = 0.01
2. LSTM units = 64, dropout = 0.6, embedding dim = 300 and L2 regularizer = 0.1
3. LSTM units = 16, dropout = 0.2, embedding dim = 300 and L2 regularizer = 0.001

And optimal configurations are used to for final run and results are mentioned in the result section. All the configurations are checked by plotting the graph of loss and accuracy of the model, results are plotted in appendix section for the most optimal model.

4.1 Hyper-parameter tuning

GridSearch is used for hyperparameter tuning, which is the process of selecting the optimal hyperparameters for a machine learning model. Hyperparameters are parameters that are not learned during the training process, but rather set before training begins, and they can significantly affect the performance of the model. All applied techniques to tune the model are mentioned in the scripts of all applied models.

4.2 Evaluation Metrics

The evaluation metrics are used to measure the quality of predictions from the classification algorithms as the data is quite imbalanced. The main classification metrics used to evaluate are precision/recall/f-score, accuracy and confusion matrix.

5 Result

As we already talked about baseline setup according to median relevance which we took 60 percent. Now we can compare the results for all these traditional and advanced techniques for prediction of search result relevance. For traditional methods we will see different techniques to check how the models react for median relevance based on precision, recall and f1 score as accuracy of all these models are around 62 percent. As precision provides us the information that how precise is the model to predict right/positive, how many of them are actually right/positive.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

[8] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

[9] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

[10] https://scikit-learn.org/stable/modules/ensemble.html

Recall on the hand is a very important measure if the cost of false negatives is very high. But in our case, we can rely on precision as it is the most important factor for us to compare relevance prediction study.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

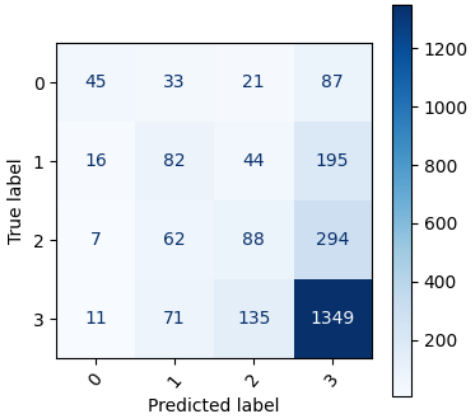
And for F1 score it is a function of precision and recall. We normally use it when we want a balance between precision and recall. Though precision is our most important factor.

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

All the best parameters and best score calculated on test data is also attached in the given scripts. Evaluation metrics and confusion matrices are mentioned below.

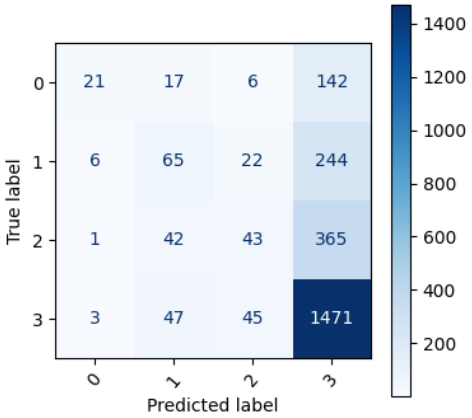
5.1 Naive Bayes

Classification	Precision	Recall	F1 score
1	0.57	0.24	0.34
2	0.33	0.24	0.28
3	0.31	0.20	0.24
4	0.70	0.86	0.77



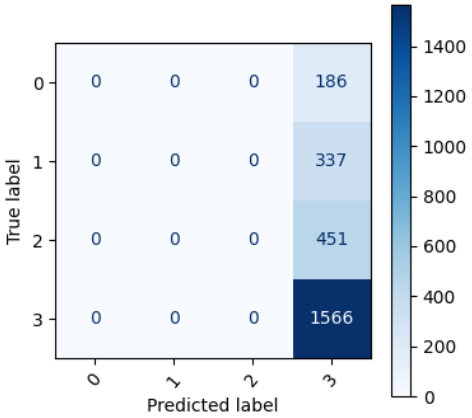
5.2 SVM

Classification	Precision	Recall	F1 score
1	0.68	0.11	0.19
2	0.38	0.19	0.26
3	0.37	0.10	0.15
4	0.66	0.94	0.78



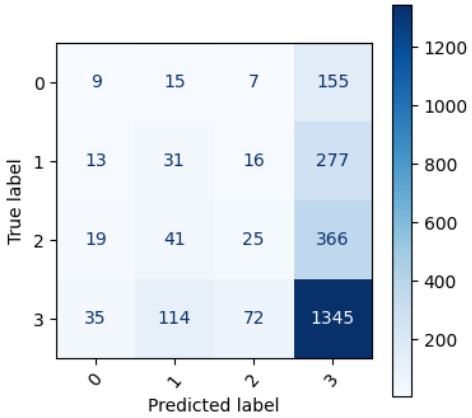
5.3 Random Forest

Classification	Precision	Recall	F1 score
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.62	1.00	0.76



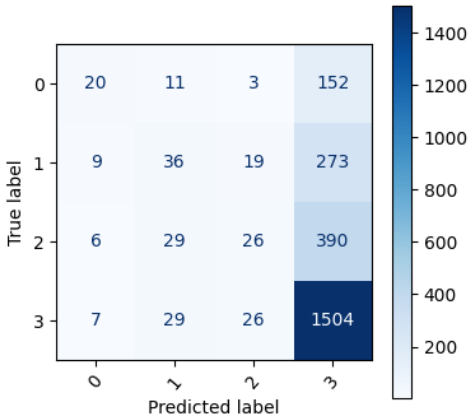
5.4 MLP

Classification	Precision	Recall	F1 score
1	0.12	0.05	0.07
2	0.15	0.09	0.12
3	0.21	0.06	0.09
4	0.63	0.86	0.73



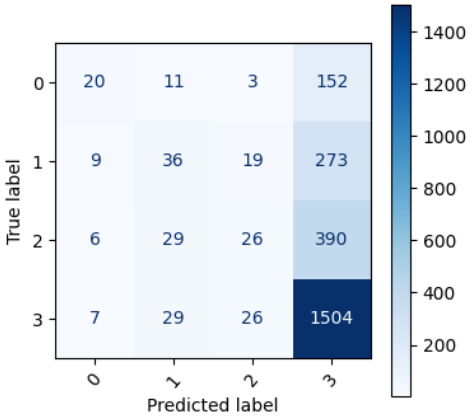
5.5 Extra-Trees (ET)

Classification	Precision	Recall	F1 score
1	0.48	0.11	0.18
2	0.34	0.11	0.16
3	0.35	0.06	0.10
4	0.65	0.96	0.77



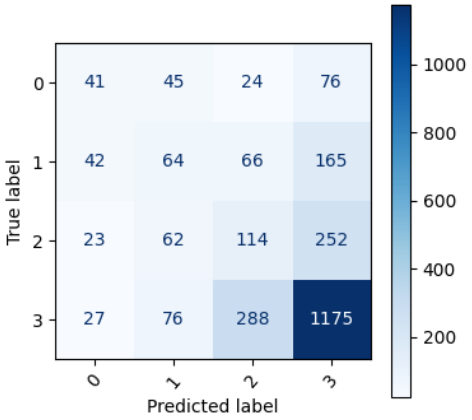
5.6 Elastic-Net

Classification	Precision	Recall	F1 score
1	0.48	0.11	0.18
2	0.34	0.11	0.16
3	0.35	0.06	0.10
4	0.65	0.96	0.77



5.7 BiDirectional Long Short-Term Memory(BiLSTM)

Classification	Precision	Recall	F1 score
1	0.31	0.22	0.26
2	0.26	0.19	0.22
3	0.23	0.15	0.24
4	0.70	0.75	0.73



6 Discussion

First of all, we will focus on traditional methods applied to predict relevance of search results, we set the baseline to 60 percent described before. We got accuracy of 62 percent from all the traditional methods. That's why we chose a different technique to check which method performed well.

We choose to go with precision, recall and F1 score and for us precision is the most important measure described before in the results section. We can see Naïve Bayes performing really well as the precision of median relevance = 4 is 70 percent compared to other methods where we get 66 percent with SVM and 62 percent with Random Forest. As mentioned before Random Forest performed

worst in all these methods because the precision score for other classes is 0 percent which makes sense as the data is very biased for class 4.

It can be seen that unbiased data is a limitation and may be with biased data results will be very different. But having high numbers for class 4 is much better which proves that the search algorithm is quite good. Another possibility which proves very handy is to clean the data as much as possible, the cleaner data we have the better the models work. After using normal text mining techniques like removing stop words, special symbols and numeric data we were still missing a few specific words which are useless for the model so they were added separately in the stop word dictionary to have better results. It's really hard to find these kinds of words from the data. It proves that more understanding and exploration of data is needed. And more machine power is also required as for gridsearch in most of cases computations took more then 2 hours with simple settings as the embedding layer is big for the available system.

For the bad performance of advanced methods data cleaning can be an issue and also size of data which is always a barrier for deep neural methods. More study of data can be proved handy to assign proper weights to classes. Which needs more time and exploration of data but in networking training required a much bigger model to handle long text string features and with two of those features the number of embedding layers were big too, as in BiLSTM number of parameters went more then 6,415,764 and others configurations are mentioned in appendix section.

7 Related work

In this thesis project(Martin, 2017) Mobyen Uddin Ahmed investigates the performance of search engine that uses NLP methods in order to find the use of triplets could improve search results. In this paper(Nogueira et al., 2019) Rodrigo Nogueira, Wei Yang, Jimmy Lin and Kyunghyun Cho proposed a method to improve the information retrieval effectiveness of a search engine by expanding documents based on neural networks. Also, a lot of work is done in this Kaggle competition[11] the goal was to develop an open-source model that measure the relevance of search results to help small business owners to compete against resource rich competitors.

8 Conclusion

Though deep learning is very popular and is a breakthrough. But still traditional methods also proved to be very handy, especially when we have limited data and limited resources. This whole implementation and research prove that it's not a better approach to dig too deep when the cost of action is higher and outcome is not much. Also, measuring performance of models shouldn't only rely on accuracy of the model specially when dealing with highly biased data. At the end Exploratory Data Analysis (EDA) plays the most important role to train a better model. BiLSTM no doubt proves to be the best option but with the cost of resources and with the limited system for this research it's not a good practice to use it. But if there is no system limitation then BiLSTM can do amazing job for even a very biased dataset.

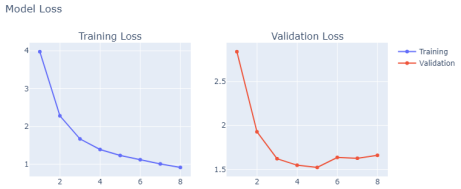
9 Appendix

9.1 BiLSTM network settings

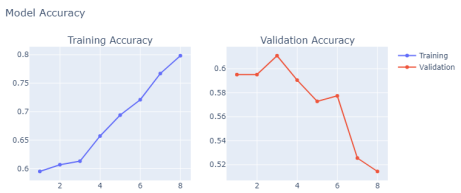
Layer (type)	Output Shape	Param #	Connected to
input_11 (InputLayer)	[(None, 300)]	0	[]
input_12 (InputLayer)	[(None, 300)]	0	[]
embedding_7 (Embedding)	(None, 300, 300)	6330000	['input_11[0][0]', 'input_12[0][0]']
bidirectional_7 (Bidirectional)	(None, 64)	85248	['embedding_7[0][0]', 'embedding_7[1][0]']
concatenate_7 (Concatenate)	(None, 128)	0	['bidirectional_7[0][0]', 'bidirectional_7[1][0]']
dense_7 (Dense)	(None, 4)	516	['concatenate_7[0][0]']

Total params: 6,415,764
Trainable params: 6,415,764
Non-trainable params: 0

9.2 BiLSTM model loss



9.3 BiLSTM model Accuracy



[11] https://www.kaggle.com/competitions/crowdflower-search-relevance/overview

References

Zabit Hameed and Begonya Garcia-Zapirain. 2020. Sentiment classification using a single-layered bilstm model. *Ieee Access*, 8:73992–74001.

Carlstedt Martin. 2017. Using nlp and context for improved search result in specialized search engines.

Rodrigo Frassetto Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *CoRR*, abs/1904.08375.