A Syntax Aware BERT for Identifying Well-Formed Queries in a Curriculum Framework

Avinash Madasu  
Department of Computer Science  
University of North Carolina at Chapel Hill  
Email: avinashm@cs.unc.edu

Anvesh Rao Vijjini  
Department of Computer Science  
University of North Carolina at Chapel Hill  
Email: anvesh@cs.unc.edu

Abstract—A well formed query is defined as a query which is formulated in the manner of an inquiry, and with correct interrogatives, spelling and grammar. While identifying well formed queries is an important task, few works have attempted to address it. In this paper we propose transformer based language model - Bidirectional Encoder Representations from Transformers (BERT) to this task. We further imbibe BERT with parts-of-speech information inspired from earlier works. Furthermore, we also train the model in multiple curriculum settings for improvement in performance. Curriculum Learning over the task is experimented with Baby Steps and One Pass techniques. Proposed architecture performs exceedingly well on the task. The best approach achieves accuracy of 83.93%, outperforming previous state-of-the-art at 75.0% and reaching close to the approximate human upper bound of 88.4%.

I. INTRODUCTION

Search Engines like Google and Bing, and E-commerce websites like Amazon and Flipkart rely on well-formed user queries to retrieve results expected by the user. However, query usage can vary from user to user though having the same intent. This can be understood as every user has a particular usage of words, grammatical knowledge and its structure [1]. Though a query can be formed in different ways, its intent remains the same making it difficult for the search engines to understand. The query language can deviate from Natural Language (NL) grammatical rules such as queries made of only nouns, pronouns and verbs, spelling mistakes. This makes standard Natural Language Processing (NLP) tools trained on well-formed data difficult to understand these queries. Furthermore, inherent sensitiveness of the language can create ambivalent semantic outputs for the same query debasing user experience.

Identifying if a query is well formed helps in restructuring queries better understood by the search engines. Early attempts were made to classify a query into informational and non-informational using supervised and unsupervised methods [2]. Segmenting a query into sub-parts using nouns improved precision of document retrieval [3]. As the time progressed, queries became much more sophisticated hence requiring advanced NLP techniques and datasets. In this direction forward, a new annotated dataset is introduced with natural language questions and the probabilities of being well-formed [4].

Transformer based language models achieved State-of-the-art (SotA) performance on Natural Language Understanding (NLU) tasks. However, a significant breakthrough was achieved by BERT [5], a Bi-directional Transformer Encoder pretrained on large amounts of data and fine-tuned for downstream NLU tasks. Since then, BERT is being used to perform fine-tuning on other tasks such as Domain Adaptation [6], Question Answering [7].

Curriculum learning introduced and formulated by [8] is a method in which neural networks are trained using easy samples at first and in the succeeding phases, difficult samples are introduced to the model. Its inspiration lies in cognitive science theories [9], [10] which propose for humans to acquire a skill, they are first provided easier variants of the task following increased difficulty. Effectiveness of Curriculum Learning has been explored in Natural Language Understanding tasks such as Question Answering [11] and Natural Answer Generation [12] and more recently in the broad range of GLUE tasks as well [13]. [14] proposed Baby Steps and One Pass curriculum techniques using sentence length as a curriculum strategy for training LSTM [15] on Sentiment Analysis. A tree-structured curriculum ordering based on semantic similarity is proposed by [16]. [17] propose a auxiliary network which is first trained on the dataset and used to calculate difficulty scores for the curriculum ordering. Some of these works [14], [16], [17] have suggested that Baby Steps performs better than One Pass. We perform experiments using both the techniques. While the idea of curriculum remains same across these works, the strategy itself to decide sample ordering is often tough to decide. In this work, we exploit the query well-formedness probability scores given in the query well-formedness dataset [4] to design our curriculum ordering. In doing so we forego the experiments required to choose a curriculum ordering and furthermore, build a strategy of difficulty which agrees with the perspective of annotators and domain experts.

The overall contributions of our paper are as follows:

- We establish a new state-of-the-art performance in query well-formedness identification utilising BERT.
- Proposed framework utilizes query well-formedness probability scores to determine a curriculum ordering for training. Samples are ordered in this strategy based on how difficult is it to differentiate them between well-formed or poorly-formed.
- Proposed architecture also infuses syntactic formation in the form of Parts-of-Speech in a manner that helps the
model achieves better performance.

In Section II, we explain previous SotA approaches which tackle Query well-formedness identification and also, curriculum learning for text classification works. Section III explains the query well-formedness dataset. Section IV explains our Approach. Section V provides our architecture in a detailed manner including the curriculum algorithm. Section VI lists down the baselines for comparison and other recent architectures proposed for this task. Section VII explains our results and Section VIII concludes our work.

II. RELATED WORK

While the importance of query well-formedness prediction is immense, few works have addressed it. Faruqui & Das [4] propose solutions for the task using character, word and parts-of-speech features. They propose a feed forward neural network with two hidden layers which takes word, character and POS n-gram from SyntaxNet POS tagger [18], all concatenated into the input layer. Their approach shows a significant improvement by including POS features.

Faruqui & Das [4] further identify that a query is often not well formed because of its ungrammatical nature. POS sequences often encode syntactic information and hence help in identifying ungrammatical structures in poorly formed queries.

Syed [19] proposed the popular ULMFiT [20] architecture for Query well formedness. They employ a framework of Inductive Transfer Learning to train their architecture. In this framework, the ULMFiT architecture which consists of stacked AWD-LSTMs [21] is first pretrained in a language model setting on open domain data. It is followed by further language model fine tuning on the query well-formedness dataset and finally the actual classification down stream task. Language model pre-training helps the model learn and understand the nature, semantics and structure of regular language. As a consequence, when the model is tasked with differentiating between grammatical and ungrammatical construct, it is able to utilize the language model information.

Furthermore, as [4] suggest, well formedness of a query relies heavily on grammatical structure and how semantically sound the query is. This parallels the dataset of Corpus of Linguistic Acceptability CoLA [22]. CoLA is a dataset intended to gauge at the linguistic competence of models by making them judge the grammatical acceptability of a sentence. CoLA is also part of the popular GLUE benchmark datasets for Natural Language Understanding [23]. The transformer [24] encoder architecture BERT [5] is known for outperforming previous GLUE SotAs, including CoLA. Hence we utilise BERT as our base architecture.

III. DATASET

To evaluate our proposed model, we perform experiments on query well-formedness dataset [4]. This dataset has collection of user generated queries and their associated annotator tagged query well-formedness scores. For the dataset building process, annotators consider a query well-formed if the words are spelled properly, the sentence syntax is syntactically and grammatically sound and if the sentence is interrogative in the pragmatic and discourse sense. We use the same train, dev and test splits and 0.8 is the threshold used for classification. Unless specified, the results reported from our experiments are averaged across 5 runs. Example samples from this dataset along with their associated well-formedness scores can be observed in Table I.

IV. PROPOSED APPROACH

In this section, we describe architecture of the proposed model. The architecture of the proposed model is shown in figure 2.

A. BERT

Bidirectional Encoder Representations from Transformers (BERT) [5] is a masked language model trained on a large corpora. A sentence is added with a special token (CLS) at the beginning and is passed into pretrained BERT model. It tokenizes the sentence with a maximum length of 512 and outputs a contextual representation for each of the tokenized words. There are variants of pretrained BERT depending upon hyper-parameters of the model. BERT-Base Uncased consists of 12 transformer encoders and output from each token is a 768 dimension embedding. BERT-Large Uncased is a stack of 24 transformer encoders and the output from each token is of 1024 dimension.

B. POS-LSTM

Although BERT is capable of learning underlying grammatical structure [25], modelling their structures can significantly improve performance. Infusing BERT with syntactic information in terms of Parts-of-Speech tags is done in previous works as well, such as [26]. However, they align POS information with BERT time steps. This leads to certain POS tags to be segmented since BERT relies on WordPiece tokenization [27]. WordPiece segments out-of-vocabulary words into subwords that are observed in vocabulary. Aligning POS tags to WordPiece segments will affect the POS n-gram information negatively. Hence, we employ a separate LSTM trained on only POS tags thereby keeping their sequence and structure independent of BERT’s tokenization. Significant improvement in LSTM performance by modelling them on syntactic information has been observed by previous works such as [28].

C. Effective utilization of query well-formedness probability for curriculum learning

The significance of the probability score for a query was often overlooked by the previous literature. They considered 0.8 as the threshold and performed binary classification of the query being well-formed or not. It is intrinsically assumed that queries belonging to a particular class have the same characteristics. However, this assumption is often not true. For example, assume queries $q_1$, $q_2$, $q_3$ and $q_4$ with probability scores 0.0, 0.6, 0.8 and 1.0 respectively. $q_1$ and $q_2$ belong to

1We use the NLTK toolkit for POS tagging: https://www.nltk.org/book/ch05.html
### TABLE I: Examples of Difficult and Easy samples according to the proposed curriculum.

| Label | Example | Query well-formedness probability |
|-------|---------|----------------------------------|
| Easy  | 1. 1961 penney worth ? | 0.0 |
|       | 2. How many years of college do you have to do to be an airline pilot ? | 1.0 |
|       | 3. 8OZ IN 1000ML ? | 0.0 |
| Medium | 1. Why use nautical mile unit ? | 1.0 |
|       | 2. Where is Ann Bates grave ? | 0.2 |
|       | 3. What is turkey population ? | 0.2 |
| Hard  | 1. How many potatoes is 350 grams ? | 0.8 |
|       | 2. Can hammerhead sharks eat ? | 0.4 |
|       | 3. What is the worlds easiest drawing ? | 0.6 |

(a) The distribution of the annotated question according to well-formedness probability as per \[4\] and our curriculum sets described.

![Graph showing distribution of annotated questions](image)

(b) One Pass in Curriculum Ordered training

![Graph showing curriculum phases](image)

---

### TABLE II: Dataset Details.

| Difficulty Set | Positive | Negative | Total |
|----------------|----------|----------|-------|
| Hard           | 2578     | 2002     | 4580  |
| Medium         | 4189     | 4860     | 9049  |
| Easy           | 3189     | 3773     | 7962  |

same class whereas \(q_3\) and \(q_4\) belong to different class. \(q_1\) is easy to distinguish from \(q_4\) because they are distant from threshold and hence are easy to train. However, \(q_2\) and \(q_3\) are hard to train because they are close to threshold and are quite difficult to distinguish. Therefore, we categorize training samples into hard, medium and easy based on their scores. The category division is shown in Figure 1a. We employ curriculum learning to train these categories so that adequate importance is given to hard samples.

### V. ARCHITECTURE DETAILS

Let \(X\) be the sentence where \(X = \{x_1, x_2, ... , x_m\}\), where \(m\) is the sentence length and varies for each sentence.

#### A. BERT sub-network

Sentence \(X\) is passed through BERT pretrained model which gives output from the last layer:

\[
H^L_M = BERT(X)
\]  

where \(M\) is the maximum token length of the BERT and \(L\) is the number of encoder layers in BERT. The context embedding from [CLS] token in the last layer is \(H^L_0\).

#### B. POS sub-network

Let \(P\) be the Parts-of-Speech tags for the sentence \(X\) where \(P = \{p_1, p_2, ..., p_m\}\) and \(m\) is the sentence length. The embeddings for POS tags are initialized randomly and are updated during training. Let \(E\) be the embedding matrix for POS tags \(P\).

\[
E = emb(P)
\]  

This embedding vector is trained using LSTM through which grammatical structure is modelled. Let \(h_t\) be the output from the final time-step of LSTM.

\[
h_t = LSTM(E)
\]  

The output from LSTM is passed through a fully connected layer.

\[
C = f(h_t \cdot W_c)
\]  

where \(f\) is the activation function.
Fig. 2: Proposed POS-BERT Architecture

C. Dense sub-network

The outputs $H_L^b$ and $C$ from BERT sub-network and POS sub-network respectively are concatenated.

$$T = [H_L^b, C]$$  \hfill (5)

The concatenated output $T$ is passed onto a fully connected layer.

$$S = f(T \cdot W_t)$$  \hfill (6)

The output $S$ from the fully connected layer is passed onto fully connected softmax layer.

$$O = g(S \cdot W_s)$$  \hfill (7)

where $g$ is the softmax activation function.

D. Hyper-parameters

We consider 44 as the maximum sentence length for BERT tokenization and 20 as the maximum sentence length for POS tags. 768 is the context embedding dimension for BERT Base Uncased whereas 1024 is the size of context embedding for BERT Large Uncased. ReLU is the activation function used in all the fully connected layers except in final layer. A dropout of 0.5 is applied on the concatenated output. Batchsize used is 50 and the model is trained for 10 epochs. AdamW is used as the optimizer with a learning rate of 2e-5 and epsilon value 1e-8. We selected the model that performed best on validation data. Early Stopping with a patience of 5 is used if the model accuracy is not improving on validation data.

E. Curriculum Training

We propose to use two curriculum methods for training our model: One Pass and Baby steps. They are described in Algorithms 1 and 2 respectively. Here $f_w$ is the classification model, and $D$ is the whole Dataset made up of sets based on difficulty $D_1, D_2$ and $D_3$. In Baby Steps, we first train the model using easy samples. In the second phase, model is trained with both easy and medium samples. In the final phase, model is trained with the entire dataset. In One Pass, model is trained using hard, easy and medium samples subsequently. Note that in one pass samples which used to train in the previous phase are not added to the samples used in the current phase.

Algorithm 1 One Pass Curriculum

1: procedure OP-CURRICULUM($f_w, D$)
2: Obtain Ordering $(D_1, D_2, D_3)$
3: for $D_i = (D_1, D_2, D_3)$ do
4: Train $f_w$ on $D_i$
5: end for
6: Test $f_w$
7: end procedure

Algorithm 2 Baby Steps Curriculum

1: procedure BS-CURRICULUM($f_w, D$)
2: Obtain Ordering $(D_1, D_2, D_3)$
3: $D = D_1$
4: Train $f_w$ on $D$
5: for $D_i = (D_2, D_3)$ do
6: $D = D + D_i$
7: Train $f_w$ on $D$
8: end for
9: Test $f_w$
10: end procedure

VI. BASELINES

We compare the proposed model to previous State-of-the-Art (SotA) architectures.

A. Question Word Classifier

If the query in the test set begins with a question classify it as well-formed [19].
B. Majority Class Prediction

In this model, we classify all the queries into the majority class of the test set [19].

C. Word BiLSTM

A Bidirectional LSTM is trained with one-hot vectors as the input from the words of a query. The output from final time-step is passed onto a softmax classifier to perform binary classification [19].

D. word-1,2 char-3,4 grams

A feed-forward neural network is trained with word-1,2 grams and character-3,4 grams as input features. All the features are concatenated to form the input to the network. Every feature is represented as a real-valued embedding [i].

E. word-1,2 POS-1,2,3 grams

In this model, in addition to word-1,2 grams, POS features are extracted using SyntaxNet POS tagger [18]. All these features are concatenated to form input to the feed-forward neural network.

F. word-1,2 char-3,4 POS-1,2,3 grams

This model’s input is a concatenation of features such as word-1,2 grams, character-3,4 grams and POS-1,2,3 grams trained using feed-forward neural network.

G. Inductive Transfer Learning (ITL)

In this model, Pretrained Language Model ULMFiT [20] is fine-tuned onto the well-formedness query dataset. The Fine-tuned Language Model is used to train a classifier by adding a softmax layer on the top of ULMFiT Language Model [19].

VII. RESULTS AND DISCUSSION

Table III captures the results of proposed approach with SotA approaches along with ablation studies for the proposed approach in terms of Accuracy and F1 score. As we see in Table III the data is imbalanced, in such cases F1 is a much more precise metric for comparison. However previous approaches have only measured Accuracy for the task, and we utilize the same for comparison with these works.

A. BERT-Base outperforms previous works

Even without any additional inputs, we observe that BERT-Base Uncased and BERT-Large Uncased outperforms previous approaches. Both the BERT methods and ITL are language models pretrained on huge amounts of open domain data which infuse them with general language structure. However, the self-attention characteristic of BERT helps it to attend to positions which cause the sentence to be poorly formed such as incorrect spellings or ungrammatical structures thereby improving performance with such identifications. To further investigate the importance of self-attention the task of query well-formedness identification we visualize BERT’s attention in Figure 3. Firstly, we notice that BERT’s attention heads attend to different ideas relevant for the classification task. These ideas especially include incorrect spellings and ungrammatical structures. Works such as [31] establish that BERT’s heads attend on linguistic notions of syntax. We find that BERT is able to identify and attend to incorrect spellings exceptionally well, even for a case where there are multiple incorrect spellings in a sentence (“develpement” and “enlightenment”). Previous approaches [4, 19] would treat an incorrect spelling as simply an out-of-vocabulary word. WordPiece tokenizer allows it to break incorrect spellings and attend to improbable sub-word sequences helping in predicting poorly formed queries. We found certain heads focusing on syntax and grammar as well. An example is visualized where BERT is attending on a noun phrase with incorrect number information (“interiors designer”).

B. Importance of POS level knowledge

We see that the introduction of POS LSTM to BERT, further improves the result. This reinforces the idea that parts-of-speech information is critical in capturing unsound grammar [4]. However, the improvement over the introduction of POS to BERT is less than the introduction of POS-level information to BERT’s architecture. This shows that while POS helps in the task, BERT alone also has good knowledge of syntax and grammar, an idea widely popular in previous works as well [25].

C. Curriculum Strategies

Finally, among One Pass and Baby Steps curriculum strategies, we see that both perform exceedingly well, with results very close to approximate human upper bound as defined by Faruqui & Das [4]. One Pass is more efficient since it only observes a sample once, unlike Baby Steps which repeatedly sees samples from previous phases along with current phase’s samples. In our experiments we observe that both the approaches perform the best under different orderings of samples. Baby Steps performed best in an Easy to Hard setting like in a typical curriculum. However, one pass performed best with Hard first, followed by Easy to Medium setting. Previous works [14], [16], [17] have shown that just an easy first approach does not work for One Pass Strategies. In the proposed curriculum setting difficulty of samples is defined based on the well formedness probability scores. The intuition behind categorizing samples as difficult is whether an annotator would get confused in distinguishing the sample between well formed or poorly formed. This can be observed in Table I which shows examples of Easy and Hard samples according to the proposed difficulty strategy. It’s fairly easy to distinguish between the well formed and poorly formed queries in the Easy samples and conversely for the Hard subset. Interestingly, in Figure 1 we see that such a definition of difficulty agrees with the proposed approach as well, since we see that in One Pass 1st phase, training on hard samples leads to performance as bad as majority class prediction. This implies that the proposed BERT model is approaching the task of query well-formedness in a human like manner which explains the near human level performance. The accuracy scores for one pass in each run are as follows: 83.7, 83.85,
### Table III: Comparison of proposed approach with recent architectures. FT denotes Fine-Tuning

| Model                                      | Acc | F1   |
|--------------------------------------------|-----|------|
| Question Word Classifier [19]              | 54.9| -    |
| Majority Class Prediction [19]             | 61.5| -    |
| Word BiLSTM Classifier [19]                | 65.8| -    |
| word-1,2 POS, 1-3,4 grams [4]              | 70.7| -    |
| word-1,2 char, 3-4 POS-1,2,3 grams [4]     | 70.2| -    |
| ITL [19]                                   | 75.0| -    |
| BERT-Base                                  | 78.56| 77.4 |
| BERT-Large                                 | 82.55| 82   |
| BERT-Large + POS                           | 83.04| 82.2 |
| BERT-Large + POS + Baby Steps Curriculum   | 83.39| 83   |
| BERT-Large + POS + One Pass Curriculum     | 83.93| 83.5 |
| Approx. human upper bound [4]               | 88.4| -    |

| 'give at least two different descriptions of relative location?' |  |  |
| 'What is drawing insights?' |  |  |
| 'Who is governor in Indiana?' |  |  |
| 'What beliefs were not the bases for development of the enlightenment though?' |  |  |
| 'Who is the most famous gymnast dancer?' |  |  |
| 'How much would interiors designer get per year?' |  |  |

Fig. 3: BERT’s Attention visualized. All visualizations but the last one are from the $10^{th}$ head of the first layer. The last attention is from the $12^{th}$ head of the same layer.

VIII. CONCLUSION

We propose BERT for the task of identifying well formed queries. Visualizations of BERT’s attentions further shed light behind its effectiveness. We further enrich our performance by utilizing Parts-of-Speech information in proposed architecture. POS tags are passed through a LSTM to learn POS sequences which capture grammatical information relevant for the task.

Finally, we use the query well formedness probability scores to derive a curriculum order for training the proposed approach. The curriculum learning is done in one-pass and baby steps setting in orders that suit them best. Proposed approach achieves near human performance and outperforms previous approaches.

REFERENCES

[1] C. Barr, R. Jones, and M. Regelson, “The linguistic structure of English web-search queries,” in Proceedings of the 2008 Conference on
Empirical Methods in Natural Language Processing, Honolulu, Hawaii: Association for Computational Linguistics, Oct. 2008, pp. 1021–1030. [Online]. Available: https://www.aclweb.org/anthology/D08-1107

[2] R. Baeeza-Yates, L. Calderón-Benavides, and C. González-Caro, “The intention behind web queries,” in International symposium on string processing and information retrieval. Springer, 2006, pp. 98–109.

[3] S. Bergsma and Q. J. Wang, “Learning noun phrase segmentation,” in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 819–826.

[4] M. Faruqui and D. Das, “Identifying well-formed natural language questions,” in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 798–803.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[6] X. Ma, P. Xu, Z. Wang, R. Nallapati, and B. Xiang, “Domain adaptation with BERT-based domain classification and data selection,” in Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 76–83. [Online]. Available: https://www.aclweb.org/anthology/D19-6109

[7] R. Osuna, N. El-Makky, and M. Torki, “Question answering using hierarchical attention on top of BERT features,” in Proceedings of the 2nd Workshop on Machine Reading for Question Answering. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 191–195. [Online]. Available: https://www.aclweb.org/anthology/D19-5825

[8] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 41–48.

[9] B. F. Skinner, “Reinforcement today.” American Psychologist, vol. 13, no. 3, p. 94, 1958.

[10] K. A. Krueger and P. Dayan, “Flexible shaping: How learning in small steps helps,” Cognition, vol. 110, no. 3, pp. 380–394, 2009.

[11] M. Sachan and E. Xing, “Easy questions first? a case study on curriculum learning for question answering,” in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2016, pp. 453–463.

[12] C. Liu, S. He, K. Liu, and J. Zhao, “Curriculum learning for natural answer generation.” in IJCAI, 2018, pp. 4223–4229.

[13] B. Xu, L. Zhang, Z. Mao, Q. Wang, H. Xie, and Y. Zhang, “Curriculum learning for natural language understanding,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6095–6104.

[14] V. Cirik, E. Hovy, and L.-P. Morency, “Visualizing and understanding curriculum learning for long short-term memory networks,” arXiv preprint arXiv:1611.06204, 2016.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] S. Han and S.-H. Maeng, “Tree-structured curriculum learning based on semantic similarity of text,” in 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2017, pp. 971–976.

[17] V. A. Rao, K. Anuranjana, and R. Mamidi, “A sentiwordnet strategy for curriculum learning in sentiment analysis,” arXiv preprint arXiv:2005.04749, 2020.

[18] C. Alberti, D. Weiss, G. Coppola, and S. Petrov, “Improved transition-based parsing and tagging with neural networks,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1354–1359.

[19] B. Syed, V. Indurthi, M. Gupta, M. Srivastava, and V. Varma, “Inductive transfer learning for detection of well-formed natural language search queries,” in European Conference on Information Retrieval. Springer, 2019, pp. 45–52.

[20] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” arXiv preprint arXiv:1802.05751, 2018.

[21] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and optimizing lstm language models,” arXiv preprint arXiv:1708.02182, 2017.

[22] A. Warstadt, A. Singh, and S. R. Bowman, “Neural network acceptability judgments,” Transactions of the Association for Computational Linguistics, vol. 7, pp. 625–641, 2019.