DuQM: A Chinese Dataset of Linguistically Perturbed Natural Questions for Evaluating the Robustness of Question Matching Models

Hongyu Zhu[♠♣†], **Yan Chen**[♣†], **Jing Yan**[♣†], **Jing Liu**[♣*], **Yu Hong**[♠*], **Ying Chen**[♣], **Hua Wu**[♣], **Haifeng Wang**[♣]

[♠] School of Computer Science and Technology, Soochow University, China

[♣] Baidu Inc., Beijing, China

{hines.zhu, tianxianer}@gmail.com

{chenyan22, yanjing09, liujing46, chenying04, wu_hua, wanghaifeng}@baidu.com

Abstract

In this paper, we focus on the robustness evaluation of Chinese Question Matching (QM) models. Most of the previous work on analyzing robustness issues focus on just one or a few types of artificial adversarial examples. Instead, we argue that a comprehensive evaluation should be conducted on natural texts, which takes into account the fine-grained linguistic capabilities of QM models. For this purpose, we create a Chinese dataset namely DuQM which contains natural questions with linguistic perturbations to evaluate the robustness of QM models. DuQM contains 3 categories and 13 subcategories with 32 linguistic perturbations. The extensive experiments demonstrate that DuQM has a better ability to distinguish different models. Importantly, the detailed breakdown of evaluation by the linguistic phenomenon in DuQM helps us easily diagnose the strength and weakness of different models. Additionally, our experiment results show that the effect of artificial adversarial examples does not work on natural texts. Our baseline codes and a leaderboard are now publicly available.[1]

1 Introduction

The task of *Question Matching (QM)* aims to identify the question pairs that have the same meaning, and it has been widely used in many applications, e.g., community question answering and intelligent customer services, etc. Though neural QM models have shown compelling performance on various datasets, including Quora Question Pairs (QQP) (Iyer et al., 2017), LCQMC (Liu et al., 2018), BQ (Chen et al., 2018) and AFQMC[2], neural models are often not robust to adversarial examples, which means that the neural models predict unexpected outputs given just a small perturbations on the inputs. As the example 1 in Tab. 1 shows, a model might not distinguish the minor difference (*"面 noodles"*) between the two sentences, and thus predicts the two questions semantically equivalent.

Recently, it attracts a lot of attentions from the research community to deal with the robustness issues of neural models on various NLP tasks, such as question matching, natural language inference and machine reading comprehension. Early works examine the robustness of neural models by creating certain types of artificial adversarial examples (Jia and Liang, 2017; Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2020), and involving human-and-model-in-the-loop to create dynamic adversarial examples (Nie et al., 2020; Wallace et al., 2019). Further studies discover that a few types of superficial cues (i.e. shortcuts) in the training data, are learned by the models and hence affect the model robustness (Gururangan et al., 2018; McCoy et al., 2019; Lai et al., 2021). Besides, several studies try to improve the robustness of the neural models by adversarial data augmentation (Min et al., 2020) and data filtering (Bras et al., 2020). All these efforts motivate us to better find and fix the robustness issues.

However, there are several limitations in previous studies. Firstly, the analysis and evaluation in previous work focus on just one or a few types of adversarial examples or shortcuts, but we need normative evaluation (Linzen, 2020; Ettinger, 2020; Phang et al.). In the normative evaluation (Linzen, 2020), the objective is not to fool the system by exploiting its particular weaknesses, but to comprehensively evaluate the basic linguistic capabilities of the models with a variety of systemically controlled datasets. Checklist (Ribeiro et al., 2020), QuAIL (Rogers et al., 2020) and Textflint (Wang et al., 2021) are great attempts of normative evalua-

[†]Equal contribution. The work was done when Hongyu Zhu was doing internship at Baidu.

[*]Co-corresponding authors.

[1]Code: https://github.com/baidu/DuReader/tree/master/DuQM; leaderboard: https://aistudio.baidu.com/aistudio/competition/detail/116/0/introduction.

[2]It is from Ant Technology Exploration Conference (ATEC) Developer competition, which is not available now.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7782–7794

tions. However, it is not clear that if the artificial adversarial training is effective on natural texts from real-world applications (Morris et al., 2020). Some other works manually perturb the examples to construct natural examples, but the manual perturbation is time consuming and costly (Gardner et al., 2020). Moreover, to the best of our knowledge, there are few Chinese datasets for QM robustness evaluation.

Towards this end, we create an open-domain Chinese dataset namely **DuQM** containing natural questions with linguistic perturbation for evaluating the robustness of QM models. (1) By *linguistic*, we mean that the dataset provides a detailed breakdown of evaluation by linguistic phenomenon. As shown in Tab. 1, there are 3 categories and 13 subcategories with 32 linguistic perturbations in DuQM, which enables us to evaluate the model performance by each category instead of just a single metric. (2) By *natural*, we mean all the questions in DuQM are natural, that are issued by the users in Baidu search. This design can help us to properly evaluate model's robustness on natural texts rather than artificial texts, which may not preserve semantics and the distribution of which is far from real-world applications.

The contributions of this paper can be summarized as follows:

- We construct a Chinese dataset namely DuQM that contains linguistically perturbed natural questions from Baidu search. It is a systemically controlled dataset to test various linguistic capabilities of the models (see Sec. 2). Additionally, except for few categories, most of the categories' construction methods can be easily extended to other languages (see Sec. 3).
- Our experimental results reveal three characteristics of DuQM: (1) DuQM is challenging, and has better discrimination power to distinguish the models that perform comparably on other datasets (see Sec. 4.2). (2) The detailed breakdown of evaluation by linguistic phenomena in DuQM helps diagnose the advantages and disadvantages of different models (see Sec. 4.3). (3) The whole DuQM dataset composed of natural examples. Our experimental result shows that the artificial adversarial training fails in natural texts of DuQM. DuQM can help us properly evaluate the models' robustness (see Sec. 4.4).

The remaining of this paper is organized as follows. Sec. 2 describes the 3 categories and 13 subcategories with 32 linguistic perturbations in DuQM. Sec. 3 gives the construction process of DuQM. In Sec. 4, we conduct experiments to demonstrate 3 characteristics of DuQM. We conclude our work in Sec. 5.

2 Linguistic Perturbations in DuQM

The design of DuQM is aimed at normative evaluation, that contains a detailed breakdown of evaluation by linguistic phenomenon. Hence, we create DuQM by introducing a set of linguistic features that we believe are important for model diagnosis in terms of linguistic capabilities. Basically, 3 categories of linguistic features are used to build DuQM, i.e., lexical features (see Sec. 2.1), syntactic features (see Sec. 2.2), and pragmatic features (see Sec. 2.3). We list 3 categories, 13 subcategories with 32 operations of perturbation in Tab. 1. The detailed descriptions of all categories are given in this section.

2.1 Lexical Features

Lexical features are associated with vocabulary items, i.e. words. As a word is the smallest independent but meaningful unit of speech , an operation on a single word may change the meaning of the entire sentence. It is a basic but crucial capability for models to understand word and perceive word-level perturbations. To provide a fine-grained evaluation for model's capability of lexical understanding, we further consider 6 subcategories:

Part of Speech. Parts of speech (POS), or word classes, describe the part a word plays in a sentence. DuQM considers 6 POS in Chinese grammar, including noun, verb, adjective, adverb, numeral and quantifier, which are content words that carry most meaning of a sentence. In this subcategory, we aim to test the models' understanding of words with different POSs by replacing them with related but not identical words. As the example 1 in Tab. 1 shows, inserting only one noun "面 *noodles*" makes the sentence meaning different. Furthermore, in this subcategory we provides a set of examples focusing on phrase-level perturbations to check model's capability on understanding word groups that act collectively as a single part of speech(see Exp. 11).

Named Entity. Different from common nouns that refer to generic things, a named entity (NE) is a proper noun which refers to a specific real-world object. The close relation to world knowledge makes NE ideal for observing models' under-

Category	Subcategory	Perturbation Operation	Label #Y / #N	BERT base	ERNIE base	RoBERTa base	MacBERT base	RoBERTa large	MacBERT large	Examples and Translation
Lexical Feature	Part of Speech	insert n.	-/539	41.4±3.4	40.8±2.1	43.0±0.7	41.4±2.5	**45.4±4.1**	37.3±2.4	**E1**: 鸡蛋怎么炒好吃 / 鸡蛋 面 怎么炒好吃 how to fry eggs / how to fry egg noodles
		insert v.	-/131	39.4±0.4	33.8±2.6	37.4±2.0	35.9±2.7	**39.9±3.1**	29.5±3.8	**E2**: 梦到西红柿意味着什么 / 梦到 摘 西红柿意味着什么 what does it mean to dream of tomatoes / what does it mean to dream of picking tomatoes
		insert adj.	-/458	23.5±1.9	19.2±3.7	**26.9±4.4**	23.9±4.2	18.1±2.4	10.4±2.1	**E3**: 有哪些类型的app / 有哪些类型的 移动 app what are types of apps / what are types of mobile apps
		insert adv.	-/302	3.7±0.5	4.2±0.5	3.8±0.6	4.4±1.2	**5.8±1.5**	3.1±1.1	**E4**: 为什么打嗝 / 为什么 老 打嗝 why burp / why always burp
		replace n.	-/702	86.6±0.3	86.7±0.1	88.3±0.3	88.8±1.2	**89.4±1.6**	87.8±0.7	**E5**: 申请美国 绿卡 流程是什么 / 申请美国 签证 流程是什么 what is U.S. green card application process / what is U.S. visa application process
		replace v.	-/466	71.7±1.1	77.6±0.8	76.9±0.4	76.5±1.2	81.0±1.6	**81.5±2.2**	**E6**: 为什么 下蹲 膝盖疼 / 为什么 下跪 膝盖疼 why knee pain when squatting / why knee pain when kneeling
		replace adj.	-/472	74.3±2.1	80.0±1.0	77.6±0.7	81.6±0.5	**82.7±1.1**	82.7±1.6	**E7**: 耳朵出血 严重 吗 / 耳朵出血 正常 吗 is the ear bleeding serious / is the ear bleeding normal
		replace adv.	-/188	19.1±6.1	19.3±4.4	16.3±3.8	23.9±4.6	**59.0±4.0**	56.2±2.0	**E8**: 为什么会 经常 头晕 / 为什么会 有点 头晕 why regularly feel dizzy / why slightly feel dizzy
		replace num.	-/1116	83.2±1.4	91.4±0.4	85.9±1.8	87.2±0.9	88.1±0.5	**91.9±1.1**	**E9**: 血压 130 /100高吗 / 血压 120 /100高吗 is blood pressure 130 /100 high / is blood pressure 120 /100 high
		replace quantifier	-/22	30.3±6.9	25.7±5.2	33.3±2.6	**34.9±2.6**	27.3±0.0	34.8±10.5	**E10**: 一 束 花多少钱 / 一 枝 花多少钱 how much is a bunch of flower / how much is a flower
		replace phrases	-/197	98.0±0.0	**98.1±0.2**	96.6±0.3	97.8±0.5	97.8±0.2	97.5±0	**E11**: 如何 提高自己的记忆力 / 如何 增加自己的实力 how to improve my memory / how to increase my strength
	Named Entity	replace loc.	-/458	**96.0±0.6**	95.7±0.2	95.4±0.4	95.0±0.4	94.7±0.4	94.5±0.5	**E12**: 山西 春节习俗是什么 / 陕西 春节习俗是什么 what is Shanxi spring festival customs / what is Shannxi spring festival customs
		replace org.	-/264	**94.9±0.2**	94.3±0.6	91.2±1.4	93.4±0.7	93.5±0.3	93.8±0.1	**E13**: 北京邮电大学 附近酒店有哪些 / 南京邮电大学 附近酒店有哪些 what are hotels near BUPT / what are hotels near NJUPT
		replace person	-/468	90.3±1.3	91.0±0.9	88.7±1.6	91.4±1.6	92.3±1.3	**93.2±1.1**	**E14**: 陈龙 的妻子是谁 / 成龙 的妻子是谁 who is Long Chen 's wife / who is Jackie Chan 's wife
		replace product	-/170	83.7±2.6	88.2±2.1	82.4±6.9	83.3±0.3	86.0±1.7	**88.8±4.4**	**E15**: iphone 6 多少钱 / iphone6x 多少钱 how much is iphone 6 / how much is iphone6x
	Synonym	replace n.	405/-	51.1±1.1	59.7±1.3	59.7±2.2	60.7±2.0	63.3±3.1	**71.6±4.0**	**E16**: 猕猴桃 的功效是什么 / 奇异果 的功效是什么 what are the health benefits of Chinese gooseberry / what are the health benefits of Kiwi
		replace v.	372/-	80.0±0.9	81.1±1.6	82.5±0.0	83.2±1.2	84.0±2.0	**88.1±1.4**	**E17**: 什么果汁可以 减肥 / 什么果汁可以 减重 what juice can lose weight / what juice can slim
		replace adj.	453/-	75.7±1.3	77.3±1.1	78.8±2.5	74.8±0.5	79.4±3.4	**88.5±1.3**	**E18**: 有趣 搞笑的广告词有哪些 / 幽默 搞笑的广告词有哪些 what are the funny advertising words / what are the humorous advertising words
		replace adv.	26/-	98.7±2.1	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100±0.0**	**100.0±0.0**	**E19**: 总是 想睡觉是为什么 / 老是 想睡觉是为什么 why always want to sleep / why repeatedly want to sleep
	Antonym	replace adj.	-/305	50.6±3.4	69.6±2.9	65.0±1.5	73.1±4.3	**91.7±2.3**	90.7±2.3	**E20**: 什么水果脂肪 低 / 什么水果脂肪 高 what fruit is low in fat / what fruit is high in fat
	Negation	negate v.	-/153	69.9±9.6	88.9±1.3	84.8±2.9	**93.3±1.3**	88.4±0.9	91.4±3.4	**E21**: 为什么宝宝哭 / 为什么宝宝 不 哭 why baby cries / why baby doesn't cry
		negate adj.	-/139	73.1±8.5	84.2±1.2	82.7±1.4	88.0±1.5	88.0±2.9	**89.4±1.0**	**E22**: 为什么苹果是红的 / 为什么苹果 不是 红的 why apple is red / why apple is not red
		neg.+antonym	59/-	29.9±2.5	34.4±2.5	39.0±1.7	31.1±2.5	40.7±1.7	**53.6±0.9**	**E23**: 激动 怎么办 / 无法 平静 怎么办 what to do if too excited / what to do if can't calm down
	Temporal word	insert	-/120	26.6±2.1	29.1±2.1	33.1±0.9	41.7±3.3	**47.5±5.4**	33.6±8.5	**E24**: 北京会下雨吗 / 北京 明天 会下雨吗 will it rain in Beijing / will it rain in Beijing tomorrow
		replace	-/114	44.1±6.1	67.8±2.6	55.0±0.5	53.8±1.3	70.4±6.1	**78.6±5.8**	**E25**: 昨天 下雪 了 吗 / 明儿 会下雪吗 was it snow yesterday / will it snow tomorrow
Syntactic Feature	Symmetry	swap	533/-	97.3±0.4	**98.0±0.1**	95.2±1.7	95.9±0.7	93.3±0.9	92.5±1.9	**E26**: 鱼 和 鸡蛋 能一起吃吗 / 鸡蛋 和 鱼 能一起吃吗 can I eat fish with egg / can I eat egg with fish
	Asymmetry	swap	-/497	14.5±2.0	18.3±3.7	26.8±3.2	26.4±2.5	**52.0±4.6**	49.1±10.8	**E27**: 北京 到 上海 航班有哪些 / 上海 到 北京 航班有哪些 what are the flights from Beijing to Shanghai / what are the flights from Shanghai to Beijing
	Negative Asymmetry	swap + negate	49/-	**47.6±3.4**	37.4±7.7	44.2±1.1	25.8±3.1	23.1±6.7	29.9±1.9	**E28**: 男人 比 女人 更 高 吗 / 女人 比 男人 更 矮 吗 are men taller than women / are women shorter than men
	Voice	insert passive word	94/37	76.8±1.4	72.5±0.0	77.4±0.9	74.0±0.7	**85.2±1.4**	74.8±2.2	**E29**: 梦见狗咬左腿意味着什么 / 梦见 被 狗咬左腿意味着什么 what does it mean to dream of being bitten by a dog / what does it mean to dream of being bitten by a dog
Pragmatic Feature	Misspelling	replace	468/-	**68.0±2.0**	65.1±0.2	64.2±0.6	65.0±2.3	63.5±1.8	63.2±1.6	**E30**: 什么 纹身 适合我 / 什么 文身 适合我 what tattoo suits me / what tatoo suits me
	Discourse Particle (Simple)	insert or replace	213/-	98.7±0.5	98.4±0.2	98.6±0.5	99.2±0.2	99.5±0.0	**99.8±0.2**	**E31**: 人为什么做梦 / 那么 人为什么做梦 why people dream / so why people dream
	Discourse Particle (Complex)	insert or replace	131/-	46.5±0.6	56.2±2.0	64.1±2.0	61.6±1.6	65.1±3.4	**68.4±0.3**	**E32**: 附近最好的餐厅有哪些 / 求助我旁边 哪家餐厅 最好吃 ? what is the best restaurant nearby / heeelp!!! which restaurant is best in my area ?
Total	**13**	**32**	**2803/7318**	-						-

Table 1: Categories of DuQM (described in Sec. 2) and performance of 6 models on each subcategory (discussed in Sec. 4). **Bold face** and underlined indicate the first and second highest accuracy for each testing scenario.

standing of the meaning of names and background knowledge about entities. Thus, we include *Named Entity* as an independent subcategory to test the model's behavior of named entity recognition, and focus on 4 types of NE most commonly seen, i.e., location, organization, person and product. Example 12 is a search query and its perturbation on NE. The two named entities, *"*山西 *Shanxi"* and *"*陕西 *Shaanxi"*, are similar at character level but denote two different locations. We expect that the models can capture the subtle difference.

Synonym. A synonym is a word or phrase that means exactly or nearly the same as another word or phrase in a given language. This subcategory aims to test whether models can identify two semantically equivalent questions whose surface forms only differ in a pair of synonyms. As in example 16, the two sentences differ only in two words, both of which refer to Kiwifruit, and has the same meaning.

Antonym. In contrast to synonyms, antonyms are words within an inherently incompatible binary relationship. This subcategory examines model's capability on distinguishing words with opposed meanings. We mainly focus on adjective's opposite, e.g., "高*high*" and "低*low*" (see example 20).

Negation. Negation is another way to express contradiction. To negate a verb or an adjective in Chinese, we normally put a negative before it, e.g., "不*not*" before "哭*cry*" (example 21), "不是*not*" before "红的*red*" (example 22). The negative before the verb or the adjective negates the statement. It is an effective way to analyze model's basic skill of figuring out the contradictory meanings even there is only a minor change. Moreover, we include some equivalent paraphrases with negation in this subcategory. In example 23, "无法平静*can't calm down*" is the negative paraphrase of "激动*excited*", so that the paraphrase sentence is equivalent to the positive sentence. We believe that a robust QM system should be able to recognize this kind of paraphrase question pairs.

Temporal Word. Temporal reasoning is the relatively higher-level linguistic capability that allows the model to reason about time. Unlike English, verbs in Chinese do not have morphological inflections. Tenses and aspects are expressed either by temporal noun phrases like "明天*tomorrow*" (examples 24) or by aspect particles like "了*le*" which indicates the completion of an action (example 25). This subcategory focuses on the temporal distinctions and helps us evaluate the models' temporal reasoning capability.

2.2 Syntactic Features

While single word sense is important to question meaning, how words composed together into a whole also affects sentence understanding. We believe that the relations among words in a sentence is important for models to capture, so we focus on several types of syntactic features in this category. We pre-define 4 linguistic phenomena that we believe is meaningful to locate model's strength and weakness, and introduce them in this subsection.

Symmetry. Sometimes paraphrases can be generated by only swapping two conjuncts around in a structure of coordination. As shown in example 26, "鱼*fish*" and "鸡蛋*egg*" are joined together by the conjunction "和*and*", which have the symmetric relation to each other. Even if we swap them around, the sentence meaning will not change. We name this subcategory *Symmetry*.

Asymmetry. Some words (such as "和and") denote symmetric relations, while others (for example, preposition "到to") denote asymmetric. Example 27 shows a sentence pair in which the word before the preposition "到*to*" is an adverbial and the word after it is the object. Swapping around the adverbial and the object of the prepositional phrase will definitely leads to a nonequivalent meaning. If a model performs well only on subcategory *Symmetry* or *Asymmetry*, it may rely on shortcuts instead of the understanding of the syntactic information.

Negative Asymmetry. To further explore the syntactic capability of QM model, DuQM includes a set of test examples which consider both syntactic asymmetry and antonym, and we name this category *Negative Asymmetry*. In example 28, the asymmetric relation between "男人*men*" and "女人*women*" and the opposite meaning of "高*taller*" and "矮*shorter*" resolve to an equivalent meaning. With this subcategory, we can better explore model's capability of inferring more complex syntactic structure.

Voice. Another crucial syntactic capability of models is to differentiate active and passive voices. In Chinese, the most common way to express the passive voice is using Bei-constructions which feature an agentive case marker "被*bei*". The subject of a Bei-construction is the patient of an action, and the object of the preposition "被*bei*" is the agent. Compared to Fig.2(a) (in Appendix A), the additional

"被*bei*" and the change of word order of "猫*cat*" and "狗*dog*" in Fig.2(b) convert the sentence from active to passive voice, but the two sentences have the same meaning. If we further change the word order from Fig.2(b) to Fig.2(c), the sentence still uses passive voice but has different meaning. Moreover, passive voice is not always expressed with "被*bei*". Sometimes a sentence without any passive marker is still in passive voice. In example 29, although the first sentence is without "被*bei*", it expresses the same meaning as the second one. There are a set of active-passive examples in this category, which are effective to evaluate model's performance on active and passive voices.

2.3 Pragmatic Features

Lexical items ordered by syntactic rules are not all that make a sentence mean what it means. Context, or the communicative situation that influence language use, has a part to play. We include some pragmatic features in DuQM so as to observe whether models are able to understand the contextual meaning of sentences.

Misspelling. Misspellings are quite often seen by search engines and question-answering systems, which are mostly unintentional. Models should have the capability to capture the true intention of the questions with spelling errors to ensure the robustness. In example 30, despite the misspelled word "文身*tatoo*" the two questions mean the same, In some real world situations, models should understand misspellings appropriately. For example, when users search a query but type in misspelling, a robust model will still give the correct result.

Discourse Particle. Discourse particles are words and small expressions that contribute little to the information the sentence conveys, but play some pragmatic functions such as showing politeness, drawing attention, smoothing utterance, etc. As shown in example 32, the word "求助*help*" is used to draw attention and brings no additional information to the sentence. Whether using these little words does not change the sentence meaning. It is necessary to a model to identify the semantic equivalency when such words are used.

3 Construction

We design DuQM as a *diverse* and *natural* corpus. The construction process of DuQM is divided into 4 steps and illustrated in Fig. 1. Firstly, we preprocess the source questions to obtain their linguistic knowledge, which will be used to perturb the source texts. Then we pair the source and perturbed question as an example. The examples' naturalness is reviewed manually. At last, the examples are annotated manually and DuQM is finally constructed. We will introduce the construction details in this section.

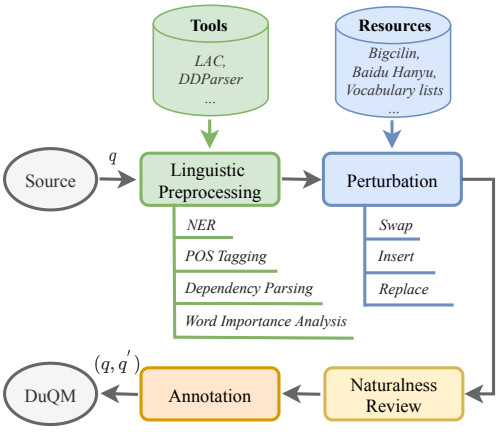


Figure 1: Construction process of DuQM.

3.1 Linguistic Preprocessing

We collect a large number of source questions from the search query log of Baidu search, and filter out question sentences with a question identification model (the accuracy is higher than 95%). All the source questions are natural that users have entered into Baidu search and then we perform several linguistic preprocessings on them: named entity recognition, POS tagging, dependency parsing, and word importance analysis. The linguistic knowledge of the source questions we obtained in this step will be used for perturbation.

3.2 Perturbation.

We conduct different perturbation operations for different subcategories. In general, we perturb the sentences in three ways:

- **replace**: replace a word with another word, e.g., for category *Synonym*, we replace one word with its synonym;
- **insert** : insert an additional word, e.g., for category *Temporal Word*, we insert temporal word to the source question;
- **swap**: swap two words. This operation is only used in *Syntactic Feature*.

The perturbation for each linguistic category is listed in column *Perturbation Operation* of Tab. 1, and the perturbation details are as follows:

Lexical Features. For each source question, we select the word with specific POS tag or entity type and high word importance score as *target word*, and perturb the source questions with some other words we collect from following 4 sources:

- Elasticsearch[3]: to collect words which have high character overlap with *target words*[4];
- Faiss[5]: to collect words which are semantically similar to *target words*; Specifically, we use RockectQA[6] to train a question dense retrieval model and employ it by faiss for similarity search;
- Bigcilin[7]: to collect synonym of *target words*;
- Baidu Hanyu[8]: to collect antonym and synonym of *target words*;
- XLM-RoBERTa (Conneau et al., 2020): to insert additional words to source sentences[9];
- Vocabulary lists[10]: to insert some specific words, such as negation word and temporal word.

Syntactic Features. For *Symmetry* and *Asymmetry*, we retrieve the source questions from the search log and select the questions whose edit distance to source question is equal to 4 as candidate questions. Then we compare the dependency structures of the source question and candidate questions. Only the question pairs which contain symmetric or asymmetric relations are retained. To generate examples for *Negative Asymmetry*, from *Asymmetry* we select the example pairs, one side of which can be negated, and negate one side of the pairs. The asymmetric syntactic structure of two sentences and one-sided negation resolves to a positive meaning. For *Voice*, we add "被*bei*" word to source questions to conduct a change of voice.

Pragmatic Features.

Misspelling. With a Chinese heteronym list[11], we obtain a set of common typos and substitute the correct-spelling words with typos. Additionally, the perturbation should satisfy two constraints: 1) the typos should be commonly used Chinese characters; 2) only one character in the source sentence is replaced with its typo.

Discourse Particle. We construct this category in 2 ways: 1) we replace or add some question words, auxiliary words or punctuation marks to generate *Simple Discourse Particle* examples (*Discourse Particle (Simple)* in Tab. 1); 2) for *Complex Discourse Particle* examples (*Discourse Particle (Complex)* in Tab. 1), we select some question pairs from a Frequently-Asked-Questions (FAQ) log, especially pairs with big differences in sentence length. Then the pairs are annotated manually and we retained the positive examples.

With above approaches, we perturb the source questions and obtain a large set of question pairs. Then the generated question pairs are manually reviewed in terms of naturalness and quality.

3.3 Naturalness Review

To ensure the generated sentences are natural, we examine their appearances in the search log and only retain the sentences which have been entered into Baidu search. Then the source question and generated question are paired together as an example.

3.4 Manual Annotation

To ensure the quality, linguistic experts from our internal data team evaluate the examples in terms of ***fluent***, ***grammatically correct***, *and* ***correctly categorized***. The low-quality examples are discarded and the examples with inappropriate categories are re-classified. Notably, examples of different subcategories are not overlapped, as we re-classified data categories and guarantee each example has one category.

Then the question pairs are annotated by the annotators[12]. Semantically equivalent question pairs are positive examples, and inequivalent pairs are negative. Each example is annotated by three annotators, and the examples will be tagged with the majority label given by the annotators. To further ensure the annotation quality, 10% of the annotated examples are selected randomly and reviewed by another senior linguistic expert and if the review accuracy is lower than 95%, the annotators need to re-annotate all the examples until

[3] https://github.com/elastic/elasticsearch

[4] Elasticsearch uses similarity ranking (relevancy) algorithm (BM25 in default) to build a search engine. Hence, we can easily to obtain high character overlap with target words with it.

[5] https://github.com/facebookresearch/faiss

[6] https://github.com/PaddlePaddle/RocketQA

[7] http://www.bigcilin.com/browser/

[8] https://hanyu.baidu.com/

[9] We add an additional $\{mask\}$ before target word, and use pre-trained language model to predict it. The prediction result of $\{mask\}$ is the word inserted to the source sentence.

[10] Vocabulary lists refer to some word lists containing specific words, such as negation word list and temporal word list.

[11] https://github.com/FreeFlyXiaoMa/pycorrector/blob/master /pycorrector/data/same_stroke.txt

[12] The annotators are linguistic experts from our internal data team

Category	Length		#		
	q	q'	Y	N	All
Lexical	8.58	8.89	1,315	6,784	8,099
Syntactic	9.86	9.89	678	532	1,210
Pragmatic	8.73	9.03	812	0	812
Avg / Total	8.74	8.90	2,805	7,316	1,0121

Table 2: Data statistics of DuQM.

Model	$\text{LCQMC}_{\text{test}}$	DuQM	△
BERT_{b}	87.1±0.1	66.6±0.6	-20.5
ERNIE_{b}	87.3±0.1	69.8±0.3	-17.5
$\text{RoBERTa}_{\text{b}}$	87.2±0.4	69.5±0.1	-17.7
$\text{MacBERT}_{\text{b}}$	87.4±0.3	70.3±0.6	-17.1
$\text{RoBERTa}_{\text{l}}$	**87.7±0.1**	**73.8±0.3**	-13.9
$\text{MacBERT}_{\text{l}}$	87.6±0.1	73.8±0.5	-13.8

Table 3: Accuracy(%) on $\text{LCQMC}_{\text{test}}$ and DuQM. $_{\text{b}}$ indicates base, and $_{\text{l}}$ indicates large.

the accuracy is higher than 95%.[13] Generally, only 0.002% (20/10,167) generated examples are not fluent or not grammatically correct, and only 0.018% (185/10,121) generated examples are re-annotated manually. The overall annotation process is illustrated in Fig. 3 (in Appendix).

Eventually, we generate 10,121 examples for DuQM. The class distribution of all categories are given in Tab. 1. Additional data statistics are provided in Tab. 2. The construction methods are not Chinese-specific. Except for few categories (e.g. Bei-construction), most of construction methods can be easily extended to other languages.

4 Experiments

In this section, we conduct experiments to discuss three characteristics (char.) of DuQM. In Sec. 4.1, we provide the experimental setup and the evaluation metrics. In Sec. 4.2 ~4.4, we give the experimental results and discussions.

4.1 Experimental Setup

Datasets. To evaluate the robustness of QM models, we select LCQMC to train the models and evaluate the models' performance on DuQM. LCQMC is a large-scale Chinese QM corpus in ***general domain*** and the source questions are collected from Baidu Knows (a popular Chinese community question answering website) (Liu et al., 2018), which ***are similar to the search queries in form***. Specifically, we firstly train QM models on $LCQMC_{train}$. Then we choose the model with the best performance on $LCQMC_{dev}$ and report the results of the chosen models on $LCQMC_{test}$ and DuQM. Tab. 8 presents the statistics of LCQMC.

It is worth mentioning that LCQMC is in general domain and its source questions are similar to the search query, which are the form of source questions for DuQM. In other words, ***DuQM is not a out-of-domain (ood) test set of LCQMC***, so that models' low performance on DuQM could not be attributed to being ood.

Models. We choose 6 popular pre-trained models to conduct experiments: $BERT_{\text{b}}$ (Devlin et al., 2019), $ERNIE_{\text{b}}$ (Sun et al., 2019), $RoBERTa_{\text{b}}$, $RoBERTa_{\text{l}}$ (Liu et al., 2019), $MacBERT_{\text{b}}$, $MacBERT_{\text{l}}$ (Cui et al., 2020). A detailed comparison is provided in Tab. 7 (in Appendix), and the training details are described in Appendix C.1.1.

Evaluation Metrics. QM problem is normally formulated as a binary classification task. Like most classification tasks, we use accuracy to evaluate a single model's performance, which is the proportion of correct predictions among the total number of the examples. As *DuQM* is a corpus consisting of a set of linguistic categories and each category differs in size, we use the ***micro-averaged*** and the ***macro-averaged accuracy*** to compare the models' performances on DuQM, which can help us better indicate the models' ability on different categories.

4.2 Char. 1: DuQM is Challenging and with Better Discrimination Ability

Tab. 3 shows the performances of models on held-out set $\text{LCQMC}_{\text{test}}$ and our DuQM, which presents the primary characteristic of DuQM: it is challenging and can better discriminate models' abilities.

As shown in Tab. 3, all models achieve accuracy higher than 87% on $\text{LCQMC}_{\text{test}}$, but show a significant performance drop on DuQM. Column △ in Tab. 3 shows the differences between models' performances on $\text{LCQMC}_{\text{test}}$ and DuQM, which presents that the performance on DuQM is lower than on $\text{LCQMC}_{\text{test}}$ by at most 20.5%. This indicates that DuQM is more **challenging**, and we claim that a challenging dataset could better distinguish the models' performance. As shown in Tab. 3, all the models have similar performances

[13] Since all annotators are linguistic experts from our internal data team instead of crowd-sourcing, we do not need to use Inter-Annotator Agreement (IAA) to measure the annotation quality.

Models		Lexical							Syntactic	Pragmatic	DuQM
		POS	NE	Synonym	Antonym	Negation	Temporal	**Lexical**			
$BERT_b$	micro	62.1±1.1	92.3±0.5	69.5±0.4	50.6±3.4	64.4±5.9	35.1±3.3	67.2±0.7	59.1±0.4	72.6±1.6	66.6±0.6
	macro	51.9±1.5	91.2±0.7	76.4±0.6	50.6±3.4	57.6±4.4	35.5±3.3	61.4±1.2	59.1±0.7	71.1±1.1	62.0±0.9
$ERNIE_b$	micro	64.6±0.5	92.8±0.4	73.2±0.9	69.6±2.9	77.8±1.1	48.0±1.9	71.0±0.3	60.0±1.2	72.4±0.3	69.8±0.3
	macro	52.4±0.7	92.3±0.6	79.5±0.7	69.6±2.9	69.1±1.2	48.5±1.9	65.5±0.5	56.5±1.0	73.2±0.8	65.1±0.3
$RoBERTa_b$	micro	64.2±0.1	90.6±1.8	74.2±1.4	65.0±1.5	76.3±1.7	43.7±0.2	70.1±0.1	63.1±0.6	73.3±0.1	69.5±0.1
	macro	53.3±0.2	89.4±2.5	80.3±1.1	65.0±1.5	68.8±1.3	44.0±0.2	65.0±0.1	60.9±0.6	75.6±0.5	65.5±0.1
$MacBERT_b$	micro	64.8±1.1	92.0±0.7	73.3±1.1	73.1±4.3	80.7±0.5	47.6±1.3	71.2±0.7	62.1±1.0	73.4±1.5	70.3±0.6
	macro	54.2±0.9	90.7±0.6	79.7±0.5	73.1±4.6	70.7±0.1	47.7±0.2	66.3±0.2	55.5±0.7	75.2±1.1	65.8±0.1
$RoBERTa_l$	micro	**67.2±0.9**	92.5±0.3	76.0±2.1	**91.7±2.3**	80.2±0.8	**58.6±2.8**	74.1±0.3	**72.6±1.4**	73.2±1.9	**73.8±0.3**
	macro	**57.7±0.6**	91.6±0.3	81.7±1.6	**91.7±2.3**	72.3±0.6	**59.0±2.7**	70.2±0.3	**63.4±1.2**	76.0±2.0	69.8±0.2
$MacBERT_l$	micro	65.6±0.8	**93.2±0.6**	**83.2±1.6**	90.7±2.3	**84.3±1.3**	55.5±4.0	**74.4±0.4**	70.2±3.7	**73.7±1.1**	73.8±0.5
	macro	54.7±0.9	**92.6±0.9**	**87.1±1.2**	90.7±2.3	**78.1±0.9**	56.1±4.0	**70.7±0.5**	61.6±2.4	**77.1±0.6**	**70.2±0.5**

Table 4: The micro-averaged and macro-averaged accuracy are on each category of DuQM.

	PWWS	$PWWS_{nat}$	FOOLER	$FOOLER_{nat}$	$CHECKLIST_{nat}$
Train	159,503	-	64,086	-	
Test	400	200	400	200	400

Table 5: Statistics of the adversarial examples.

on $LCQMC_{test}$ (around 87%), but different performance on DuQM: the accuracy of base models differs from 66.6% to 70.3%, and the large models show higher performance (73.8%). In conclusion, DuQM shows a **better discrimination ability** to evaluate models.

4.3 Char. 2: Diagnose Model in Diverse Ways

DuQM is a corpus which has 3 linguistic categories and 13 subcategories and enables a detailed breakdown of evaluation on different linguistic phenomena. In Tab. 1, we give the performance of 6 models on all fine-grained categories of DuQM, and Tab. 4 reports the micro-averaged and macro-averaged accuracy. By comparing these results, we introduce the second characteristic of DuQM: it can diagnose the strengths and weaknesses of the models in diverse ways. Several interesting observations are noticed: (from Tab. 1 and 4):

1) *In most categories*, large models outperform base models. As the large models have more parameters and larger pre-training corpus, it is reasonable that they have better capabilities than relatively smaller models.
2) In *Named Entity*, all models show good performance (higher than 90%). Another interesting finding is that although $ERNIE_b$ is a relatively small model, it performs slightly better than $RoBERTa_l$ on this subcategory, which might attribute to the entity masking strategy during pre-training.
3) $MacBERT_l$ is significantly better than other models on *Synonym*. We suppose that it benefits from its pre-training strategy that using similar words instead of random words for masking. Moreover, $RoBERTa_l$ and $MacBERT_l$ have remarkable better performance on *Antonym*.
4) Low performance in *Temporal word* show that all models lack the ability of temporal reasoning.
5) All models have surprisingly poor performance on *Asymmetry* while good performance on *Symmetry*. We suppose that lack of learning word orders would result in a wrong prediction when the words orders are altered.
6) $BERT_b$ and $ERNIE_b$ perform better on *Misspelling*, and $RoBERTa_b$ and $MacBERT_b$ are relatively better on *Complex Discourse Particles*.

In general, DuQM diagnoses models from a linguistic perspective and can help us identify the strengths and weaknesses of the models.

4.4 Char. 3: Natural Examples

DuQM is composed of adversarial testing examples generated by linguistically perturbed natural questions[14]. We consider that natural examples can better evaluate models' *robustness* than artificial examples. To demonstrate it, we conduct an experiment to compare the performance of two adversarial training (AT) methods PWWS (Ren et al., 2019) and TextFooler (Jin et al., 2020) on artificial and natural test examples:

[14] An adversarial example is an input to a machine learning model that is purposely designed to cause a model to make a mistake in its predictions despite resembling a valid input to a human. As they are designed to evaluate different linguistic capabilities of the modes, all examples in DuQM are adversarial examples.

Training set	LCQMC	Attack test set				CHECKLIST$_{nat}$	DuQM	
		PWWS	PWWS$_{nat}$	FOOLER	FOOLER$_{nat}$		Micro	Macro
LCQMC	87.7	58.1	81.5	57.1	87.8	76.9	73.8	69.8
LCQMC+PWWS	87.7$_{+0.0}$	**97.6**$_{+39.5}$	81.8$_{+0.3}$	73.1$_{+16.0}$	87.6$_{-0.2}$	76.0$_{-0.9}$	**75.2** $_{+1.4}$	70.4$_{+0.6}$
LCQMC+FOOLER	87.5$_{-0.2}$	78.5$_{+20.4}$	**83.8**$_{+2.3}$	**80.8**$_{+23.7}$	82.0$_{-5.8}$	**79.2**$_{+2.3}$	71.4 $_{-2.4}$	68.8$_{-1.0}$

Table 6: Adversarial training results of RoBERTa$_{l}$. 'FOOLER' refers to 'TEXTFOOLER'. We use green and red subscripts to represent a higher and lower accuracy respectively.

- *Artificial test examples*, which are generated *artificially* and may not preserve semantics and introduce grammatical errors. We employ two methods PWWS and TextFooler on LCQMC$_{test}$ to generate artificial adversarial examples. These two methods generate adversarial examples by replacing words with synonyms until models are fooled.
- *Natural test examples* are texts within linguistic and semantics constraints. Our annotators from the internal data team reviewed and annotated all the generated texts with methods PWWS, TextFooler and the translated texts of Checklist dataset (Ribeiro et al., 2020), and we finally get three natural test sets, PWWS$_{nat}$, TextFooler$_{nat}$ and Checklist$_{nat}$.

Besides, we employ PWWS and TextFooler on LCQMC$_{train}$ to generate artificial adversarial training examples, which are combined with original LCQMC$_{train}$ as training data (Row *LCQMC+PWWS* and *LCQMC+FOOLER* in Tab. 6).The detailed data statistics are shown in Tab. 5. AT details are in Appendix C.2.

Evaluation with artificial and natural adversarial examples. We fine-tune RoBERTa$_{l}$ on LCQMC and the artificial adversarial examples generated by PWWS and TextFooler, and evaluate on the adversarial test sets. The results are shown in Tab. 6. Row *LCQMC* shows that only training with LCQMC$_{train}$ shows a low performance on *PWWS* and *TextFooler* (we provide a detailed analysis in Appendix C.3), and the performance on *PWWS* and *TextFooler* are significantly higher on *PWWS$_{nat}$* and *TextFooler$_{nat}$* respectively. However, if we incorporate LCQMC$_{train}$ with the examples generated by PWWS and TextFooler, the model's performance on *PWWS* and *TextFooler* increase greatly (both methods achieve an great improvement of more than 16%) , but the effects on natural examples *PWWS$_{nat}$* and *TextFooler$_{nat}$* are not significant (-5.8% ~2.3%). On the other 2 natural test sets, Checklist$_{nat}$ and DuQM, the effects of 2 adversarial methods are also not obvious (-2.4% ~2.3%).

In conclusion, the common artificial AT methods are not so effective on the natural datasets. As a corpus consisting linguistically perturbed natural questions, DuQM is beneficial to a robustness evaluation to help us mitigate models' undesirable performance in real-world applications.

5 Conclusion

In this work, we create a Chinese dataset namely **DuQM** which contains linguistically perturbed natural questions for evaluating the robustness of QM models. DuQM is designed to be fine-grained and natural. Specifically, DuQM has 3 categories and 13 subcategories with 32 linguistic perturbations. We conduct extensive experiments with DuQM and the results demonstrate that DuQM has 3 characteristics: 1) DuQM is challenging and has better discrimination ability; 2) The fine-grained design of DuQM helps to diagnose the strengths and weakness of models, and enables us to evaluate the models in diverse ways; 3) Artificial adversarial training fails in the natural texts of DuQM.

Acknowledgements

This research is supported by the National Key Research and Development Project of China (No.2018AAA0101900).

Ethical Considerations

This work presents DuQM, a diverse and natural dataset for the research community to evaluate the robustness of QM models. Data in DuQM are collected from Baidu search (we are legally authorized by this company), the details are presented in Sec. 3. Since DuQM do not have any user information, there is no privacy concerns. In addition, to ensure that the DuQM is free potential biased and toxic content, we desensitize all the instances in it. Regarding to the issue of labor compensation, all annotators are employees from our internal data team and are fairly compensated.

Limitations

Our dataset DuQM provides a new resource for evaluating the robustness of QM models. However, the categories can be further expanded to consider more behavioral capabilities of QM models, such as symmetry ($(a, b) = (b, a)$) and transitivity ($(a, c)\ if\ a = b\ and\ b = c$).

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4946–4951, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. *data. quora. com.*

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The*

Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 8018–8025. AAAI Press.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002, Online. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC:a large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Jason Phang, Angelica Chen, William Huang, and Samuel R Bowman134. Adversarially filtered evaluation sets are more challenging, but may not be fair.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

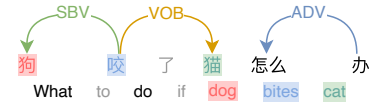
Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8722–8731.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *ArXiv preprint*, abs/1904.09223.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.



(a) Active voice question.

(b) Passive voice paraphrase question.

(c) Passive voice non-paraphrase question.

Figure 2: The dependency relations of active voice and passive voice questions.

A Dependency Relations of Bei-Construction

Fig. 2 illustrates the dependency relations of active voice and passive voice questions in Chinese.

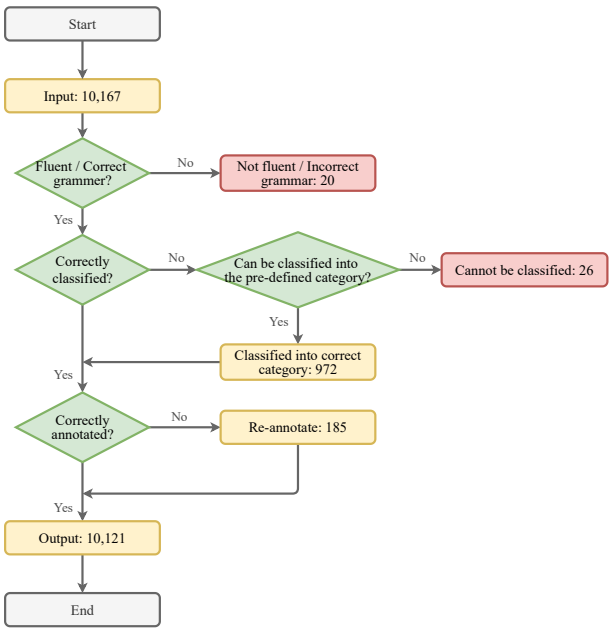


Figure 3: Overall annotation process.

B Annotation Process

Fig. 3 illustrates the overall annotation process.

C Supplementary Experiments

C.1 Additional Experimental Setting

C.1.1 Training Details

In the fine-tuning stage, we insert a $[SEP]$ between the question pairs. The pooled output is passed to a classifier. We use different learning rates and epochs for different pre-trained models. Specifically, for large models, the learning rate is 5e-6 and the number of epochs is 3. For base models, the learning rate is 2e-5, and we set the number of epochs as 2. The batch size is set as 64 and the maximal length of question pair is 64. We use early stopping to select the best checkpoint. We choose the model with the best performance on $\text{LCQMC}_{\text{dev}}$ to report test results and each model is fine-tuned 3 times on $\text{LCQMC}_{\text{train}}$.

C.1.2 Datasets Details

Tab. 8 gives a detailed description of LCQMC Corpus.

C.2 Adversarial Training Details

Tab. 5 gives a detailed statistics of adversarial examples generated with TextFooler, PAWS and Checklist. To generate training samples, we select a set of LCQMC training questions and apply the methods PWWS and TextFooler on them. The labels are same as original samples. To generate test samples and ensure a robust evaluation, we utilize 4 datasets, PWWS_{nat}, $\text{TextFooler}_{\text{nat}}$, $\text{Checklist}_{\text{nat}}$[17] and DuQM, which are natural adversarial examples. We conduct an experiment about adversarial training by feeding the models both the original data and the adversarial examples, and observe whether the original models become more robust. We use pre-trained model $\text{RoBERTa}_{\text{l}}$ (described in Tab. 7) for fine-tuning and the details are described in Sec. 4.1.

C.3 Results of Attacks

We give the main results of attacks to BERT_{b} and $\text{RoBERTa}_{\text{l}}$ in Tab. 9. The results show that the un-natural attacks (on artificial adversarial samples, i.e. PWWS and TextFooler in Tab. 9) have higher success rate than DuQM. However, if we select the natural examples from the artificial adversarial samples (PWWS_{nat} and $\text{TextFooler}_{\text{nat}}$ in Tab. 9), the attack success rate of PWWS and TextFooler is significantly decreasing by at least 18.5% on BERT_{b}

[17] Before annotating, we translate original Checklist dataset into Chinese using Baidu translate

Models	L	H	A	# of Parameters	Masking	LM Task	Corpus
BERT_b	12	768	12	110M	T	MLM	Wikipedia
ERNIE_b	12	768	12	110M	T/E/Ph	MLM	Wikipedia+Baike+Tieba, etc.
RoBERTa_b	12	768	12	110M	MLM	-	EXT[15]
MacBERT_b	12	768	12	110M	Mac	SOP	EXT
RoBERTa_l	24	1024	16	340M	MLM	-	EXT[16]
MacBERT_l	24	1024	16	340M	Mac	SOP	EXT

Table 7: The hyper-parameters of public pre-trained language models we use(L: number of layers, H: the hidden size, A: the number of self-attention heads, T: Token, E: Entity, Ph: Phrase, WWM: Whole Word Masking, NM: N-gram Masking, MLM: Masked LM, Mac: MLM as correction).

Corpus	Train	Dev	Test	Fine-grained
LCQMC	238,766	8,802	12,500	No

Table 8: Data statistics of LCQMC.

Data	BERT	RoBERTa
PWWS	41.5	41.9
PWWS_nat	23.0$_{-18.5}$	18.5$_{-23.4}$
TEXTFOOLER	**46.6**	**42.9**
$\text{TEXTFOOLER}_\text{nat}$	14.6$_{-32.0}$	12.2$_{-30.7}$
DuQM	33.4	26.2

Table 9: Attack success rate(%) on different test data.

and 30.7% on RoBERTa_l respectively. DuQM, in which all the samples are natural and grammarly correct, gets the best performance when black-box attacking (compare to PWWS_nat and $\text{TextFooler}_\text{nat}$ in Tab. 9). In summary, the artificial adversarial examples training is not effective on natural texts, such as DuQM. It is reasonable that we should pay more attention to the naturalness when generating the adversarial examples.