Controlled Cloze-test Question Generation with Surrogate Models for IRT Assessment

Anonymous ACL submission

Abstract

Item difficulty plays a crucial role in adaptive testing. However, few works have focused on generating questions of varying difficulty levels, especially for multiple-choice (MC) cloze tests. We propose training pre-trained language models (PLMs) as surrogate models to enable item response theory (IRT) assessment, avoiding the need for human test subjects. We also propose two strategies to control the difficulty levels of the distractors using ranking rules to reduce invalid distractors. Experimentation on a benchmark dataset demonstrates that our proposed framework and methods can effectively control and evaluate the difficulty levels of MC cloze tests.

1 Introduction

Multiple-choice cloze tests are fill-in-the-blank questions that assess reading comprehension and overall language proficiency by requiring test takers to select the correct missing words from options. Table 1 gives an example test item consisting of a stem with a gap to fill, a key or answer, and three distractors.

Stem:			
I knelt and put my arms around the child. Then the tears came, slowly at first , but soon she was ___ her heart out against my shoulder.			
Options:			
A. crying	B. shouting	C. drawing	D. knocking
Key: A	**Distractors:** B C D		

Table 1: A question item of MC Cloze test.

MC cloze test questions have been a focus of research because they are a common question format on standardized language proficiency exams such as TOEFL, TOEIC, IELTS, and college/high school entrance exams. In this paper, we address the research questions of generating MC cloze test of different item difficulty levels.

Prior studies on cloze test question generation have concentrated largely on distractor generation, with the goal of reproducing distractors exactly matching the benchmark datasets (Chung et al. 2020; Ren et al., 2021; Chiang et al. 2022; Wang et al. 2023). Although some studies have acknowledged the benefit of having distractors with diverse difficulty levels (Yeung et al., 2019), there has been minimal investigation into generating distractors with difficulty level different from the benchmark.

Item difficulty plays a crucial role in adaptive testing. It is a parameter that determines which questions to present to a test taker and estimates their proficiency level. Therefore, the difficulty of each item should be known beforehand so appropriate questions can be selected during the test (Susanti et al. 2017). However, only a number of works have focused on generating question items of various difficulty levels, for RC questions (Gao et al. 2019a), C-test questions (Lee et al. 2019 and 2020), and MC cloze questions (Susanti et al. 2017). This research gap is largely due to the lack of a reliable metric to evaluate the item difficulty of generated questions. Most previous study relies on human test takers and human annotation for assessing the change of difficulty levels (Susantia et al. 2017, Lee et al. 2019).

Our research has two main goals: (1) We propose two strategies to generate cloze-test questions by controlling distractor difficulty, with consideration for reducing invalid distractors. (2) We address the problem of objective and efficient evaluation by using PLMs as subject surrogates to mimic Item Response Theory, bypassing the need for human test subjects. We will release our dataset and codes under CC-BY 4.0 license upon publication.

Related Research	Answer Type	Dataset	Factors to Control/Generate			Difficulty Control (Evaluation Method)	Difficulty Level
			Distractor (Selection Method)	Gap (Generation Method)	Stem		
Gao et al. 2019a	R. C.	SQuAD			√	Yes (RC system)	Item Level
Gao et al. 2019b	R. C.	RACE	√			None	
Chung et al. 2020	R. C.	RACE	√			None	
Qiu et al. 2020	R. C.	RACE	√			None	
Felice et al. 2022	Open Cloze	private		√ (Electra)		None	
Matsumori et al. 2023	Open Cloze	private		√ (gap score)		None	
Lee et al. 2019	C-test	Beiborn et al.2016		√ (prediction)		Yes (Human Subject)	Item Level
Lee et al. 2020	C-test	Beiborn et al.2016		√ (entropy)		Yes (MLP model)	Proficiency Level
Susantia et al. 2017	MC Cloze	TOEFL iBT	√ (feature-based)		√	Yes (Human subject)	Item Level
Yeung et al. 2019	MC Cloze	Chinese sentences	√ (BERT-based ranking)			None	
Ren and Zhu, 2021	MC Cloze	DGen	√ (featured-based L2R)			None	
Panda et al. 2022	MC Cloze	ESL lounge	√ (BERT-based and feature-based)			None	
Chiang et al. 2022	MC Cloze	CLOTH, DGen	√ (BERT-based and feature-based))			None	
Wang et al. 2023	MC Cloze	CLOTH, DGen	√ (Text2Text)			None	
Our Research	**MC Cloze**	**CLOTH**	**√ (BERT-based and feature-based with validity rules)**			**Yes (PLM-based IRT Assessment)**	**Item Level**

Table 2: Recent Research on Question Generation for Language Proficiency Test

2 Related Research

The language proficiency test commonly adopts cloze tests (open or multiple-choice), C-tests, and reading comprehension (RC) to assess students' language skills. Question Generation (QG) aims to create natural and human-like questions from diverse data sources. Research on MC cloze test question generation primarily focuses on tasks such as analyzing factors influencing item difficulty (Susanti et al., 2017), distractor generation (Yeung et al., 2019; Ren and Zhu, 2021; Chiang et al., 2022), and reducing invalid distractors (Zesch and Melamud, 2014; Wojatzki et al., 2016). Table 2 presents a comparative analysis of recent studies on the automatic generation of cloze test, RC, and C-test.

For MC cloze test, distractor generation algorithms aim to identify plausible but incorrect candidates for filling in blanks. Selection is based on semantic proximity to the target word, measured through methods like WordNet (Brown et al., 2005), thesauri (Smith et al., 2010), and word embeddings similarity (Guo et al., 2016; Susanti et al., 2015; Jiang and Lee, 2017). Sakaguchi et al. (2013) introduce a discriminative approach for fill-in-the-blank quiz generation for language learners. Recent studies utilize confidence scores from BERT models (Devlin et al. 2018) for ranking distractor candidates, outperforming semantic similarity methods in correlation with human judgment (Yeung et al., 2019). Ren and Zhu (2021) apply knowledge-based techniques to help generate distractor candidates. Chiang et al. (2022) suggest BERT-based methods as superior in distractor generation. Their candidate selection relies on confidence scores from pretrained language models. Wang et al. (2023) propose a Text2Text formulation using pseudo Kullback-Leibler divergence, candidate augmentation and multi-task training, enhancing performance in generating distractors that align with benchmarks.

Item difficulty is crucial in adaptive testing, yet few studies focus on generating items with diverse difficulty levels different from standard benchmark datasets. Furthermore, these works typically rely on human test-taker evaluations (Susanti et al., 2017; Lee et al., 2019). A few studies used model judgments in RC test (Gao et al., 2019) and C-test (Lee et al., 2020). Uto et al. (2023) propose difficulty-controllable neural question generation for reading comprehension using Item Response

Theory. Ehara (2018) builds an English vocabulary knowledge dataset of Japanese English-as-a-Second-Language learners using crowdsourcing to assess learner proficiency. In related research on question difficulty estimation, QA models are also proposed to estimate difficulty through item response theory (Benedetto, 2022).

3 Methodology

Our research addresses key challenges in generating MC cloze questions. We aim to produce distractors with varying difficulty using ranking rules to eliminate invalid distractors. We also propose a PLM-based IRT assessment framework to objectively evaluate item-level difficulty changes, alleviating reliance on human annotation.

As shown in Figure 1, our approach involves: (1) training PLM-based models on benchmark data to simulate test-takers; (2) designing strategies to control difficulty by selecting distractors; (3) using PLM-based surrogate models to take the modified tests and applying IRT to evaluate difficulty changes.

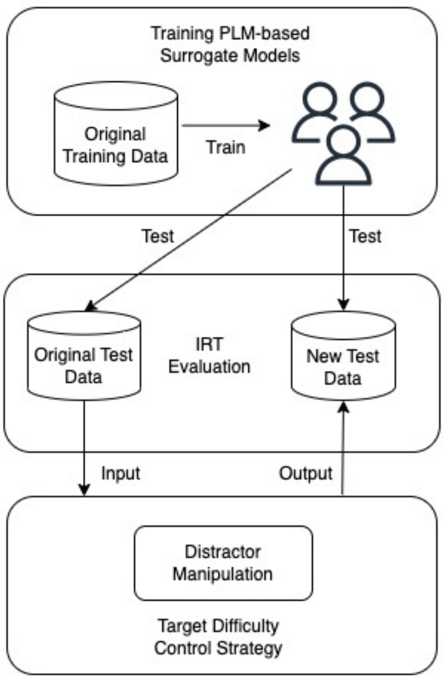

Figure 1: Research Structure

3.1 IRT Assessment with PLM-based Surrogate Models

Calibrating test difficulty traditionally requires trials with human subjects, which is time-consuming and costly. IRT is a framework to estimate item difficulty unsupervised (Benedetto, 2022; Susanti et al., 2017). Previous work used Reading Comprehension systems or MLP models to evaluate change of predictions (Gao et al. 2019a, Lee et al. 2020). We propose that predictions by various PLMs with different settings can simulate human test-taking for IRT without actual test-takers.

We fine-tune 12 PLM models on each dataset using BigBird (Zaheer et al. 2020) and Electra (Clark et al. 2020) with different hyperparameters. Control strategies generate hard and easy versions of each test fold. Trained surrogate models take these versions, and their scores are aggregated across folds. An IRT model fitted on the aggregated scores for the original and modified tests evaluates difficulty shifts between easy and hard versions by modeling score distributions.

3.2 Difficulty-controllable Question generation

For difficulty-controllable question generation, we combine PLM-based confidence scores, semantic similarity and edit distance metrics, and validity rules to generate distractors at tunable difficulty levels.

Distractor Candidate List Generation

Let:

- $V = \{v_1, v_2, \ldots, v_n\}$ be the proficiency vocabulary list, where v_i is the i-th word in the list;
- PTM be a pretrained model (e.g., BERT);
- $WP(v_i) = \{wp_1^i, wp_2^i, \ldots, wp_{m_i}^i\}$ be the set of word pieces for word v_i after tokenization using PTM, where wp_j^i is the j-th word piece of v_i and m_i is the number of word pieces for v_i.
- Q be the item question.
- $R = \{r_1, r_2, \ldots, r_k\}$ be the ranked list of the PTM word piece vocabulary, where r_i is the i-th ranked word piece and k is the total number of word pieces in the PTM vocabulary.

The algorithm to generate candidate distractors is as follows:

1. Tokenize each word $v_i \in V$ using PTM to obtain its word pieces $WP(v_i)$.
2. Predict the gap in question Q using PTM and obtain the ranked list R of the PTM word piece vocabulary.
3. For each word $v_i \in V$:

a. Calculate the rank of each word piece $wp_j^i \in WP(v_i)$ in the ranked list R. Let $rank(wp_j^i)$ denote the rank of wp_j^i.

b. Compute the mean rank of v_i's word pieces: $rank(v_i) = \frac{1}{m_i}\sum_{j=1}^{m_i} rank(wp_j^i)$

4. Sort the words in V based on their mean rank $rank(v_i)$ in ascending order to obtain the distractor candidate list $D = d_1, d_2, \dots, d_n$, where d_i is the i-th ranked word in the distractor candidate list.

Distractor Difficulty Factors:

Inspired by previous work (Susantia et al. 2017, Yeung et al. 2019, Ren and Zhu 2021, Chiang et al. 2022), we design three factors for distractor generation: semantic similarity using word2vec cosine similarity, syntactic similarity using Levenshtein distance, and PLM confidence scores for gap prediction.

- **Confidence score**

Formally, let $\mathbb{M}(\,)$ be a PLM model finetuned with our training data set, *S* be a cloze stem, *V* be a vocabulary list, *A* be the answer of *S*, and d_i be a word in *V* as a candidate distractor. We denote a given stem *S* with the cloze blank filled in $[Mask]$ with $S_{\otimes[Mask]}$.

Confidence score C_i for d_i given by PLM is defined:

$$C_i = p(d_i|\mathbb{M}(S_{\otimes[Mask]})$$

- **Semantic similarity**

The semantic similarity S_i of the candidate distractor and the answer is defined as:

$$S_i = CosineSimilarity(Embed(d_i), Embed(A))$$

where $Embed(\,)$ refers to Glove Embedding.

- **Levenshtein ratio**

The Levenshtein ratio measures string similarity on a scale from 0 to 1. It is defined as:

$$Levenshtein_{ratio} = \frac{sum - ldist}{sum}$$

where *sum* is the total length of two strings, and *ldist* is the weighted edit distance between two strings based on Levenshtein distance (Levenshtein et al. 1966). The Levenshtein distance counts insertions, deletions, and substitutions to transform one string into the other. The weighted distance is calculated as:

$$ldist = Num(INSERT) + Num(DELETE) + 2 * Num(REPLACE)$$

Distractor Selection Strategy with Validity Control

In the context of distractor selection for generating challenging or easy test items, we introduce two strategies: Confidence-Ranking Control and 3-Factor Ranking Control. These strategies aim to select distractors that are semantically similar to the correct answer while ensuring their validity.

Let $V = \{v_1, v_2, \dots, v_n\}$ denote the vocabulary list, where v_i represents the i-th word in the list. The correct answer for a given test item is denoted by a, and the set of distractors is represented by $D = \{d_1, d_2, \dots, d_m\}$, where d_i is the i-th distractor.

We define $rank_{conf}(v_i)$, $rank_{sem}(v_i)$, and $rank_{lev}(v_i)$ as the ranks of word v_i in the vocabulary list V based on BERT confidence scores, semantic similarity scores, and Levenshtein ratio scores, respectively. Additionally, let $B = \{b_1, b_2, \dots, b_l\}$ be the set of benchmark distractors, where b_i is the i-th benchmark distractor.

Previous work proposes context-sensitive lexical inference rules to filter distractors (Zesch and Melamud, 2014). Our analysis as in Appendix C reveals fine-tuned BERT often ranks invalid distractors higher than correct answers. Motivated by this and prior research (Zesch and Melamud, 2014), we design rules to reduce invalid distractors:

$$\forall d_i \in D, rank_{conf}(d_i) > rank_{conf}(a)$$

This rule guarantees that the selected distractors have lower confidence ranking positions than the correct answer.

The Confidence-Ranking Control Strategy selects distractors based on their BERT confidence scores. For hard distractors, we choose the top three distractors after the answer, as defined by:

$$D_h = \{v_i \in \mathrm{V} \mid \mathrm{rank}_{\mathrm{conf}}(a) < \mathrm{rank}_{\mathrm{conf}}(v_i) \leq \mathrm{rank}_{\mathrm{conf}}(a) + 3\}$$

For easy distractors, we select the first three distractors after the last ranked benchmark distractor, as defined by:

$$D_e = \{v_i \in V \mid \mathrm{rank}_{\mathrm{conf}}(b_l) < \mathrm{rank}_{\mathrm{conf}}(v_i) \leq \mathrm{rank}_{\mathrm{conf}}(b_l) + 3\}$$

The 3-Factor Ranking Control Strategy considers semantic similarity, Levenshtein ratio, and BERT confidence scores to select distractors.

Let K be a predefined constant that determines the range of ranks to consider after the answer or the lowest confidence-ranked benchmark distractor.

For hard distractors, we select the top two distractors by semantic similarity and the top distractor by Levenshtein ratio from the K ranks after the answer, as defined by:

$$D_{h1} = \{v_i \in V \mid \mathrm{rank}_{\mathrm{conf}}(a) < \mathrm{rank}_{\mathrm{conf}}(v_i) \leq \mathrm{rank}_{\mathrm{conf}}(a) + K,\ \mathrm{rank}_{\mathrm{sem}}(v_i) \leq 2\}$$

$$D_{h2} = \{v_i \in V \mid \mathrm{rank}_{\mathrm{conf}}(a) < \mathrm{rank}_{\mathrm{conf}}(v_i) \leq \mathrm{rank}_{\mathrm{conf}}(a) + K,\ \mathrm{rank}_{\mathrm{lev}}(v_i) = 1\}$$

$$D_h = D_{h1} \cup D_{h2}$$

For easy distractors, we select the top two distractors by semantic similarity and the top distractor by Levenshtein ratio from the K ranks after the lowest confidence-ranked benchmark distractor, as defined by:

$$D_{e1} = \{v_i \in V \mid \mathrm{rank}_{\mathrm{conf}}(b_l) < \mathrm{rank}_{\mathrm{conf}}(v_i) \leq \mathrm{rank}_{\mathrm{conf}}(b_l) + K,\ \mathrm{rank}_{\mathrm{sem}}(v_i) \leq 2\}$$

$$D_{e2} = \{v_i \in V \mid \mathrm{rank}_{\mathrm{conf}}(b_l) < \mathrm{rank}_{\mathrm{conf}}(v_i) \leq \mathrm{rank}_{\mathrm{conf}}(b_l) + K,\ \mathrm{rank}_{\mathrm{lev}}(v_i) = 1\}$$

$$D_e = D_{e1} \cup D_{e2}$$

4 Experiment Design

This section presents the experimentation details. Table 3 provides generation examples referencing the original item shown in Table 1.

4.1 Dataset

Various datasets have been used for cloze test generation (Table 1), with CLOTH[1] (Xie et al., 2017) and DGen (Ren & Zhu, 2021) being popular choices. DGen compiles science questions from diverse sources and levels, while CLOTH contains cloze-style English reading comprehension questions for middle-school and high-school entrance exams. We selected CLOTH as it aligns closely with our goal of controlling item difficulty for adaptive testing. We strictly follow the "Terms and Conditions" as listed on the download site.

We divided the CLOTH dataset into two sets according to its two proficiency levels – CLOTH-M for middle school and CLOTH-H for high school entrance exams. Each set was further segmented into 5 folds. Within each fold, we split the passages into stems. Stems comprised consecutive sentences leading up to the first [MASK] token (i.e. gap), ensuring sufficient context surrounding the cloze deletion. Data statistics is provided in Appendix A.

4.2 Evaluation

We conducted experiments on a single NVIDIA Quadro RTX 8000 GPU. The control strategies were applied to the "Test" split. By concatenating the scores across all surrogate models and test folds, IRT models were then fitted to quantify overall changes in test difficulty. We use the py-irt library (Lalor and Rodriguez, 2023) as it leverages PyTorch and GPU acceleration for faster and more scalable IRT modeling compared to existing libraries. We apply the 1PL (also known as the Rash model) with default setting. This model estimates a latent ability parameter for subjects and a latent difficulty parameter for items, which fits exactly what we intend to evaluate.

5 Results and Analysis

Surrogate Model Performance

Table 4 presents the surrogate models' average accuracies across the five test data folds on the original cloze items. Italic numbers indicate the 12 CLOTH-M surrogates' performances, while underlined numbers show the 12 CLOTH-H surrogates. The models exhibit a wide accuracy range (0.42 to 0.81), demonstrating diverse capabilities as artificial test takers for difficulty modeling.

Proficiency	CLOTH-M		CLOTH-H	
Model	BigBird	Electra	BigBird	Electra
1e-4，16	*0.4282*	*0.7106*	0.4234	0.527
1e-4，32	*0.6691*	*0.7306*	0.5671	0.6601
1e-5，16	*0.811*	*0.7613*	0.7902	0.7119
1e-5，32	*0.8081*	*0.7602*	0.7974	0.7102
3e-5，16	*0.6093*	*0.7558*	0.687	0.7008
3e-5，32	*0.798*	*0.7615*	0.7814	0.7072

Table 4. Surrogate models' performance

[1] https://www.cs.cmu.edu/~glai1/data/cloth/

	Distractor generation w/ Confidence-Ranking Control	Distractor generation w/ 3-Factor Ranking Control
Hard	(I) **Stem:** I knelt and put my arms around the child. Then the tears came, slowly at first , but soon she was ___ her heart out against my shoulder. **Options:** A. crying B. sobbing C. pouring D. weeping **Key:** A **Distractors:** B C D	(II) **Stem:** I knelt and put my arms around the child. Then the tears came, slowly at first , but soon she was ___ her heart out against my shoulder. **Options:** A. crying B. screaming C. cried D. crushed **Key:** A **Distractors:** B C D
Easy	(III) **Stem:** I knelt and put my arms around the child. Then the tears came, slowly at first , but soon she was ___ her heart out against my shoulder. **Options:** A. crying B. counting C. shouting D. booming **Key:** A **Distractors:** B C D	(IV) **Stem:** I knelt and put my arms around the child. Then the tears came, slowly at first , but soon she was ___ her heart out against my shoulder. **Options:** A. crying B. owing C. caves D. sobbed **Key:** A **Distractors:** B C D

Table 3: Generated hard and easy items for the original item in Table

To further analyze the surrogate models, we select 4 as middle school surrogates (Electra (1e-4, 16), Electra (1e-4, 32), Bigbird (1e-4, 32), and Electra (3e-5, 32)) and 3 as high school surrogates (Bigbird (1e-5, 16), Bigbird (1e-5, 32), Bigbird (3e-5, 32)). These models are trained similarly and tested on both CLOTH-M and CLOTH-H. The table below compares the average accuracies, standard deviations, and utility ratios of the 12-surrogate sets and the middle and high school surrogate subsets:

Model	CLOTH-M			CLOTH-H		
	Avg. Acc.	Stdv	Utility Ratio	Avg. Acc.	Stdv	Utility Ratio
12	0.717	0.104	73.9%	0.672	0.108	72.1%
4-mid	0.718	0.034	38.2%	0.615	0.072	52.2%
3-high	0.803	0.006	10.4%	0.79	0.007	10.9%

Table 5. Comparing surrogate models

The utility ratio is the percentage of test questions remaining after excluding those answered correctly or incorrectly by all test takers. The 4 middle school surrogates perform better on CLOTH-M and worse on CLOTH-H, while the 3 high school surrogates substantially outscore them on CLOTH-M. The smaller standard deviations demonstrate these sets represent distinct proficiency levels. The 12-surrogate sets achieve higher utility ratios (73.9%, 72.1%) than the middle and high school sets, and are retained for evaluating item difficulty control given their better utility and diverse performances to distinguish between stronger and weaker students.

Performance of Control Methods

Figures 2 and 3 present the effect of generating difficult and easy items using the Confidence-ranking algorithm and 3-Factor strategy. The red lines show IRT distributions for difficult generated items, the blue lines for easy items, and the black dotted lines mark the original test difficulty.

Both strategies systematically manipulated cloze item difficulty. Across CLOTH-M and CLOTH-H, the strategies successfully generated harder items (red distribution shift right) and easier items (blue shift left) compared to the original test items.

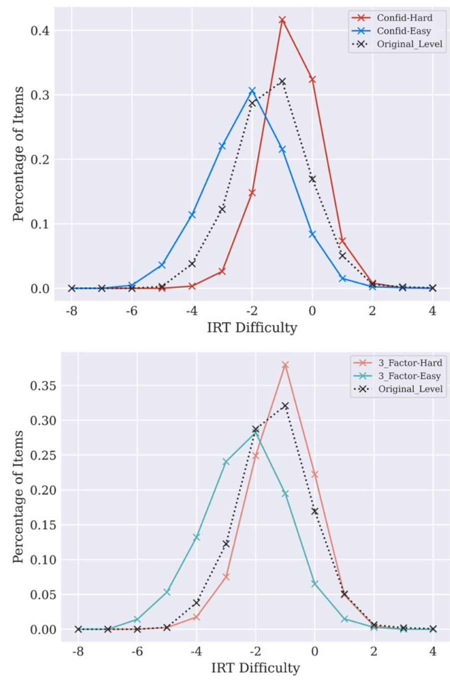

Figure 2. Change of IRT for CLOTH-M with Confidence-Ranking (above) and 3-Factor Ranking Control (below)

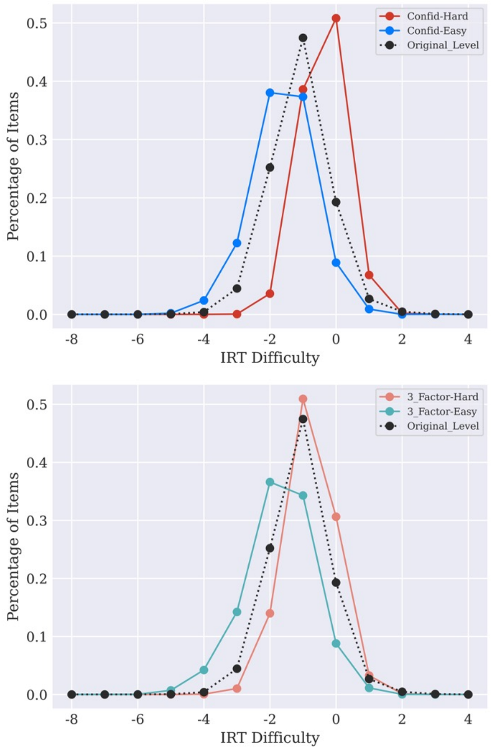

Figure 3. Change of IRT for CLOTH-H with Confidence-Ranking (above) and 3-Factor Ranking Control (below)

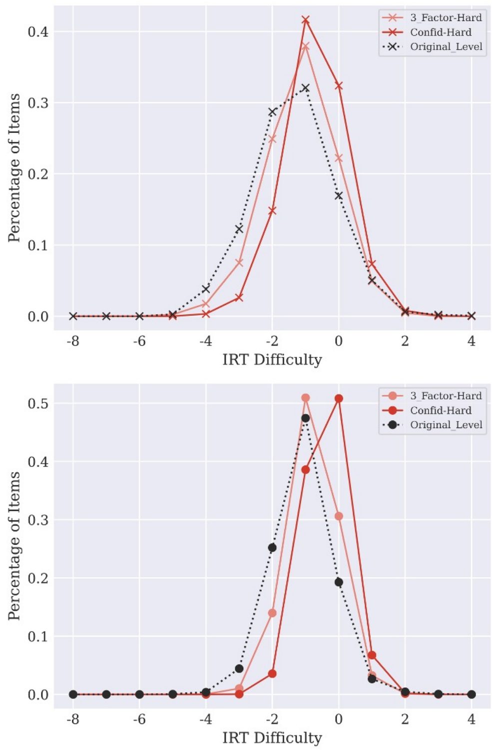

Figure 5: Confidence-Ranking and 3-Factor Ranking on generating hard items for CLOTH-M (above) and CLOTH-H (below)

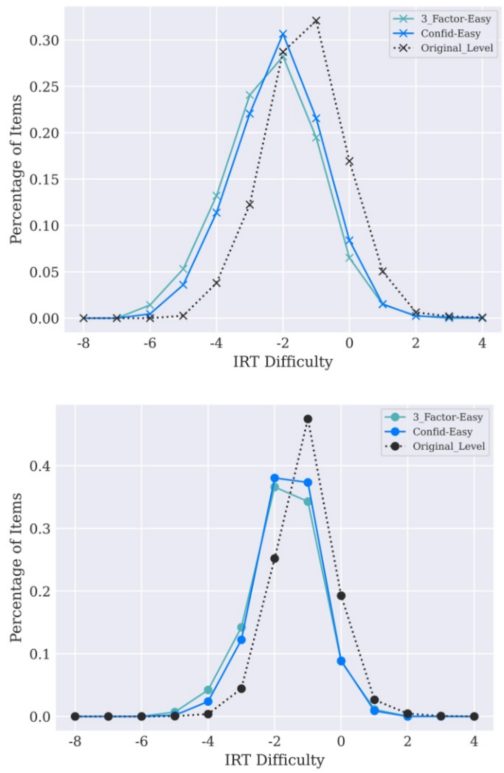

Figure 4: Confidence-ranking and 3-factor ranking on generating easy items for CLOTH-M (above) and CLOTH-H (below)

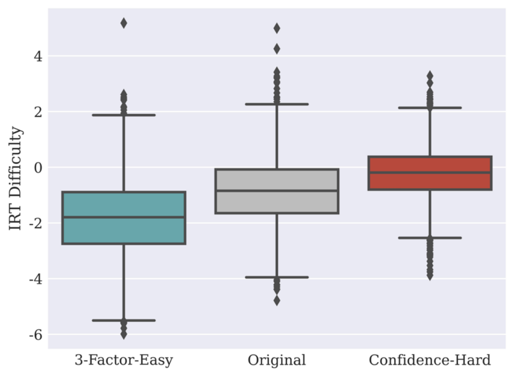

Figure 6: Best combination for CLOTH-M: 3-Factor Ranking for Easy Items and Confidence-Ranking for Hard Items Generation.

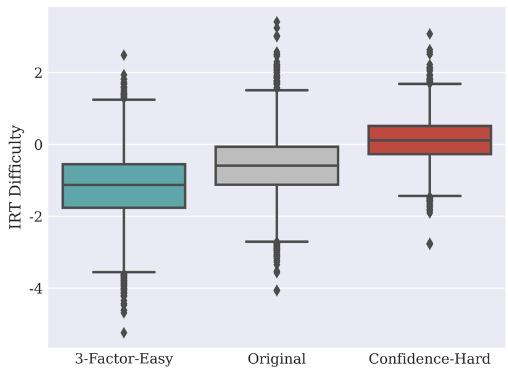

Figure 7: Best combination for CLOTH-H: 3-Factor Ranking for Easy Items and Confidence-Ranking for Hard Items Generation.

Comparing Confidence-ranking and 3-Factor Ranking on Generating Easy and Hard Items Comparing the two control strategies, 3-factor ranking generates a slightly wider range of easy item difficulties for both datasets as shown in Figure 4. Meanwhile, confidence-ranking method produces slightly wider distributions of hard items for both datasets as evident in Figure 5.

Best Combination Strategies

We provide the box plot analysis on best combination control strategies for the two proficiency tests. Figures 6 and 7 show that the best strategy combination is the 3-Factor Ranking control for easy question and confidence ranking control for hard questions

6 Human Evaluation

We hired three college instructors from the Faculty of English Language and Culture to perform a human evaluation of our generated questions. Two of them hold Ph.D. degrees in Applied Linguistics and one with two Master Degrees in Applied Linguistics and Computer-assisted Language Teaching respectively. They were compensated with a standard hourly rate for college instructors, covering four hours of training and four hours of annotation. Their payment was supervised and approved by the College Administration Office.

We randomly sampled 10 CLOTH-M articles containing 100 cloze items and 6 CLOTH-H articles containing 102 cloze items, comprising a total of 100 easy items and 102 hard items. The number of easy and hard items was equally split within both the CLOTH-M and CLOTH-H datasets. For each cloze item in each article, we randomly paired the benchmark options with the easy or hard items for the annotators to score their relative difficulty. We then curated their responses and transferred the scores as follows: the benchmark was assigned a score of 2, the higher annotated score was assigned a 3, and the lower annotation was assigned a 1. The average results are presented below.

The human evaluation results demonstrate that the generated hard cloze items are consistently perceived as more difficult than the easy items across both the CLOTH-M and CLOTH-H datasets. The total average scores for easy and hard items are 1.41 and 2.13, respectively, indicating a clear distinction in difficulty levels. This trend is also observed within each dataset, with the CLOTH-H dataset showing a more pronounced difference between easy and hard item scores compared to the CLOTH-M dataset. Despite some variations among the annotators, particularly in the CLOTH-H dataset, the overall results confirm the effectiveness of the generation process in creating cloze items with varying levels of difficulty.

Total Avg.	Ann. #1	Ann. #2	Ann. #3	Avg.
Easy	1.22	1.12	1.9	1.41
Hard	1.93	2.04	2.42	2.13

(a) Average annotation scores for the complete set

CLOTH-M	Ann. #1	Ann. #2	Ann. #3	Avg.
Easy	1.28	1.16	1.36	1.27
Hard	1.94	2.28	1.92	2.05

(b) Average annotation scores for CLOTH-M

CLOTH-H	Ann. #1	Ann. #2	Ann. #3	Avg.
Easy	1.16	1.08	2.44	1.56
Hard	1.92	1.81	2.92	2.22

(c) Average annotation scores for CLOTH-H

Table 6: Human Annotation Results

7 Conclusions

In this work, we propose a novel evaluation framework for assessing the control of item-level difficulty in multiple-choice cloze tests. By utilizing diverse pre-trained language models as surrogate test-takers, we fit Item Response Theory (IRT) distributions to quantify changes in difficulty, avoiding reliance on human subjects.

We design two strategies leveraging confidence scores, semantic similarity, and edit distance to control distractor selection for generating questions with controlled difficulty levels. To reduce the generation of invalid distractors, we implement validity rules based on prior research.

Systematic experimentation shows that (1) the advanced test (CLOTH-H) is more challenging to control than the intermediate test (CLOTH-M); (2) the 3-Factor Ranking Control method is more effective for generating easy items, while the Confidence Ranking Control method excels at generating hard items; (3) validity rules help reduce invalid distractors but do not eliminate them entirely, indicating a need for further research.

Our framework provides a promising approach for generating multiple-choice cloze questions with controllable difficulty levels, enabling more effective adaptive testing.

8 Limitation

Our study has several limitations that provide opportunities for future research. First, we acknowledge that our sample size of 12 surrogate models is smaller than conventionally recommended for Item Response Theory (IRT) analysis (Sireci, 1992). While our work explores the potential of using IRT with smaller samples, further investigation is needed to determine the optimal number of surrogate models and how well they mimic human test-takers.

Second, our study does not directly address the question of how well the surrogate models simulate actual test-takers, such as middle school and high school students taking entrance exams. Investigating this would require access to test scores from human subjects, which raises important considerations about data collection, curation, and test-taker anonymity that are beyond the scope of our current work.

Third, we recognize that IRT evaluates question difficulty based solely on test-taker responses without considering question content. Future research should explore how different parameter settings impact question quality assessment.

Finally, we opted for the 1PL (Rasch) model due to its simplicity and suitability for smaller sample sizes, as well as its alignment with our main objective of evaluating the effectiveness of our methods in controlling question difficulty. However, we acknowledge that the 2PL model may provide a more comprehensive understanding of item properties, albeit at the cost of requiring larger sample sizes for accurate parameter estimation.

Despite these limitations, we believe our work provides a valuable starting point for future research on using PLMs as surrogate test-takers and applying IRT to assess the difficulty of automatically generated questions. We hope our study will inspire further investigations into these important areas.

References

Benedetto, L. 2022. An assessment of recent techniques for question difficulty estimation from text.

Brown, J., Frishkoff, G., & Eskenazi, M. 2005. Automatic question generation for vocabulary assessment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* pages 819-826.

Chiang, S. H., Wang, S. C., & Fan, Y. C. 2022. CDGP: Automatic Cloze Distractor Generation based on Pre-trained Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5835–5840.

Chung, H. L., Chan, Y. H., & Fan, Y. C. 2020. A BERT-based distractor generation scheme with multi-tasking and negative answer training strategies. *arXiv preprint arXiv:2010.05384*.

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. 2020. Electra: Pretraining text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ehara, Y. (2018, May). Building an English vocabulary knowledge dataset of Japanese English-as-a-second-language learners using crowdsourcing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Felice, M., & Buttery, P. 2019. Entropy as a Proxy for Gap Complexity in Open Cloze Tests. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 323–327.

Felice, M., Taslimipoor, S., & Buttery, P. 2022. Constructing open cloze tests using generation and discrimination capabilities of transformers. *arXiv preprint arXiv:2204.07237*.

Gao, Y., Bing, L., Chen, W., Lyu, M. R., & King, I. 2019(a). Difficulty controllable generation of reading comprehension questions. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 4968–4974.

Gao, Y., Bing, L., Chen, W., Lyu, M. R., & King, I. 2019(b). Generating Distractors for Reading Comprehension Questions from Real Examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): pages 6423–30.

Guo, Q., Kulkarni, C., Kittur, A., Bigham, J. P., & Brunskill, E. 2016, May. Questimator: Generating knowledge assessments for arbitrary topics. In *IJCAI-16: Proceedings of the AAAI Twenty-Fifth International Joint Conference on Artificial Intelligence*.

Jiang, S., & Lee, J. S. 2017. Distractor generation for chinese fill-in-the-blank items. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* pages 143-148.

Lalor, J. P., & Rodriguez, P. 2023. py-irt: A scalable item response theory library for python. *INFORMS Journal on Computing*, *35*(1), pages 5-13.

Lee, J. U., Schwan, E., & Meyer, C. M. 2019. Manipulating the Difficulty of C-Tests. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 360–370.

Lee, J. U., Meyer, C. M., & Gurevych, I. 2020. Empowering active learning to jointly optimize system and user demands. *arXiv preprint arXiv:2005.04470*.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* Vol. 10, No. 8, pages 707-710.

Matsumori, S., Okuoka, K., Shibata, R., Inoue, M., Fukuchi, Y., & Imai, M. 2023. Mask and Cloze: Automatic Open Cloze Question Generation Using a Masked Language Model. *IEEE Access*, *11*, pages 9835-9850.

Panda, S., Gomez, F. P., Flor, M., & Rozovskaya, A. 2022. Automatic Generation of Distractors for Fill-in-the-Blank Exercises with Round-Trip Neural Machine Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 391–401.

Qiu, Z., Wu, X., & Fan, W. 2020. Automatic distractor generation for multiple choice questions in standard tests. *arXiv preprint arXiv:2011.13100*.

Ren, S., & Zhu, K. Q. 2021. Knowledge-driven distractor generation for cloze-style multiple choice questions. In *Proceedings of the AAAI conference on artificial intelligence,* Vol. 35, No. 5, pages 4339-4347

Sakaguchi, K., Arase, Y., & Komachi, M. (2013, August). Discriminative approach to fill-in-the-blank quiz generation for language learners. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 238-242).

Sireci, S.G. (1992). The Utility of IRT in Small-Sample Testing Applications.

Smith, S.; Avinesh, P.; and Kilgarriff, A. 2010. Gap-fill tests for language learners: Corpus-driven item generation. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, pages 1–6.

Susanti, Y. Iida, R. and Tokunaga, T. 2015. Automatic Generation of English Vocabulary Tests. In *Proceedings of the 7th International Conference on Computer Supported Education*, pages 77-87.

Susanti, Y., Tokunaga, T., Nishikawa, H., & Obari, H. 2017. Controlling item difficulty for automatic vocabulary question generation. *Research and practice in technology enhanced learning*, 12(1):1–16.

Uto, M., Tomikawa, Y., & Suzuki, A. (2023, July). Difficulty-controllable neural question generation for reading comprehension using item response theory. In Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023) (pp. 119-129).

Wang, H. J., Hsieh, K. Y., Yu, H. C., Tsou, J. C., Shih, Y. A., Huang, C. H., & Fan, Y. C. 2023. Distractor Generation based on Text2Text Language Models with Pseudo Kullback-Leibler Divergence Regulation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12477–12491.

Wojatzki, M., Melamud, O., & Zesch, T. 2016. Bundled Gap Filling: A New Paradigm for Unambiguous Cloze Exercises. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 172–181.

Xie, Q., Lai, G., Dai, Z., & Hovy, E. 2018. Large-scale Cloze Test Dataset Created by Teachers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2344–2356.

Yeung, C. Y., Lee, J. S., & Tsou, B. K. 2019. Difficulty-aware Distractor Generation for Gap-Fill Items. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., ... & Ahmed, A. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, pages 17283-17297.

Zesch, T., & Melamud, O. 2014. Automatic Generation of Challenging Distractors Using Context-Sensitive Inference Rules. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148.

A Data Statistics

Table 7 presents the number of items per split in our dataset.

Fold	Split	CLOTH-M	CLOTH-H
0	Train	17123	42540
	Validate	5678	14189
	Test	5669	14139
1	Train	16975	42432
	Validate	5757	14155
	Test	5738	14281
2	Train	17011	42628
	Validate	5680	14145
	Test	5779	14095
3	Train	17094	42502
	Validate	5733	14194
	Test	5643	14172
4	Train	17077	42463
	Validate	5752	14224
	Test	5641	14181

Table 7: Data Statistics

B Algorithms

Figures 8, 9 present the algorithms Confidence-Ranking Control, and 3-Factor Ranking Control respectively.

Algorithm 1: Distractor Generation with Confidence Ranking

Input : A sentence S with a cloze, Answer Word A, Pretrained Model M, Proficiency Vocabulary List V, Original Options $Options$

Output: Difficult distractors D_{hard}, Easy distractors D_{easy}

$D_{hard}, D_{easy} \leftarrow \{\}, \{\}$;
$CandidateList \leftarrow sorted(ModelPredict(S, M, V))$; `// Sort proficiency vocabulary V based on the scores predicted by the PTM M`
$OptionsRankList \leftarrow []$;
for opt *in* $Options$ **do**
  $OptionsRankList$.append($CandidateList$.index(opt));
end
$Rank_{hard} \leftarrow CandidateList$.index($A$);
$Rank_{easy} \leftarrow min(OptionsRankList)$;
$D_{hard} \leftarrow CandidateList[Rank_{hard}+1\text{: } Rank_{hard}+4]$;
$D_{easy} \leftarrow CandidateList[Rank_{easy}+1\text{: } Rank_{easy}+4]$;
return D_{hard}, D_{easy}

Figure 8: Distractor Generation with Confidence-Ranking Control

Algorithm 2: Distractor Generation with 3-Factor Ranking

Input : A sentence S with a cloze, Answer Word A, Pretrained Model M, Proficiency Vocabulary List V, Original Options $Options$, Numbers of Candidate K

Output: Difficult distractors D_{hard}, Easy distractors D_{easy}

$D_{hard}, D_{easy} \leftarrow \{\}, \{\}$;
$CandidateList \leftarrow sorted(ModelPredict(S, M, V))$; `// Sort proficiency vocabulary V based on the scores predicted by the PTM M`
$GloveSimList, LevenSimList \leftarrow [], []$;
for $word$ *in* $CandidateList$ **do**
  $G \leftarrow$ Calculate Glove similarity($A, word$);
  $L \leftarrow$ Calculate Leven similarity($A, word$);
  $GloveSimList$.append($word$, G);
  $LevenSimList$.append($word$, L);
end
$Rank_{hard} \leftarrow CandidateList$.index($A$);
$OptionsRankList \leftarrow []$;
for opt *in* $Options$ **do**
  $OptionsRankList$.append($CandidateList$.index(opt));
end
$Rank_{easy} \leftarrow min(OptionsRankList)$;
$GloveSimList_{hard}, GloveSimList_{easy} \leftarrow GloveSimList[Rank_{hard} + 1 : Rank_{hard} + K], GloveSimList[Rank_{easy} + 1 : Rank_{easy} + K]$;
$LevenSimList_{hard}, LevenSimList_{easy} \leftarrow LevenSimList[Rank_{hard} + 1 : Rank_{hard} + K], LevenSimList[Rank_{easy} + 1 : Rank_{easy} + K]$;
$D_{hard} \leftarrow$ Top 2 candidates by $GloveSimList_{hard}$, Top 1 candidate by $LevenSimList_{hard}$;
$D_{easy} \leftarrow$ Top 2 candidates by $GloveSimList_{easy}$, Top 1 candidate by $LevenSimList_{easy}$;
return D_{hard}, D_{easy}

Figure 9: Distractor Generation with 3-Factor Ranking Control

C Annotation for Invalid Distractor Control

We analyzed the issues of invalid distractors with human evaluation. We recruited 9 college students at the CET-6 English proficiency level as annotators. The annotators work with our research lab on regular basis and receive subsidy for their annotation work under supervision of our administrative office.

The invalid distractors will most likely appear when generating hard items. Using BERT's confidence score ranking without validity control, we generated distractors for 4,575 items randomly selected from the CLOTH-H dataset. Manual annotation identified 1,676 items as having at least one invalid generated distractor (i.e., a distractor that could fit as an answer in the gap). As our control strategies involves ranking distractors after the answer, we identified 302 items to further test validity rules. Among the 906 distractors generated, 482 were annotated as invalid, representing an invalidity ratio of 53.2%. After applying the Confidence-Ranking Control method and 3-Factor Ranking Control method, the ratios dropped to 20.3% and 17.3% respectively (Table 8).

Strategy	Num. of Invalid Distractors	Ratio of Invalid Distractors
Confidence ranking w/o validity rules	482	53.2%
Confidence-ranking Control	184	20.3%
3-Factor Ranking Control	160	17.7%

Table 8. Manual annotation of 906 distractors generated with confidence ranking w/o validity rules, and our methods of Confidence-Ranking Control and 3-Factor Ranking Control

The following are examples of items with the answer (bolded) and invalid distractors (italicized) generated by confidence ranking without validity rules. The same item with distractors generated using Confidence-ranking Control and 3-Factor Ranking Control is also shown below:

Example #1:

I hope I did the right thing, Mom, Alice said. I saw a cat, all bloody but alive. I [MASK] it to the vet's, and was asked to make payment immediately.

(1) Original options:
A. **carried** B. followed C. returned D. guided

(2) Distractors generated without control:
A. **carried** B. *took* C. brought D. delivered

(3) Distractors generated with 3-Factor Ranking Control:
A. **carried** B. showed C. reported D. tried

(4) Distractors generated with Confidence Ranking Control:
A. **carried** B. transported C. hauled D. rode

Example #2:

[MASK] this surprised him very much, he went through the paper twice, but was still not able to find more than one mistake, so he sent for the student to question him about his work after the exam.

(1) Original options:
A. **As** B. For C. So D. Though

(2) Distractors generated without control:
A. **As** B. *Because* C. Although D. Though

(3) Distractors generated with 3-Factor Ranking Control:
A. **As** B. Even C. Once D. Soon

(4) Distractors generated with Confidence Ranking Control:
A. **As** B. Realizing C. Again D. Initially

D Instruction to Annotators for Invalid Distractor Identification.

Instruction: You are given a set of multiple-choice cloze test questions, each with four options. The correct answer is identified, along with three generated distractor options. Please review the choices and identify any “invalid distractors” - alternatives that contextually fit the gap as a potentially correct response, rather than an implausible one.

For example:

When I began planning to move to Auckland to study, my mother was worried about a lack of jobs and cultural differences. Ignoring these ____ I got there in July 2010.

A. **concerns** B. *worries*

C. fears D. considerations

Here, the answer is "concerns". The generated distractors include "worries". Both are grammatically correct. "Concerns" fits the semantic context only slightly better. Therefore, in this case, "worries" is considered an "invalid distractor".

Your annotation results will help assess the efficacy of our difficulty-control strategies in limiting invalid distractor generation for multiple choice cloze tests.

E Instruction to Annotators for Invalid Item Difficulty Comparisons.

Instruction: You are given a set of multiple-choice cloze test questions, each with two sets of options. Please compare and mark the harder set with 2 and easy set with 1. Your annotation results will help assess the efficacy of our difficulty-control strategies for cloze item generation.

For example:

I have a good friend at school. Her name is Liu Mei. She's fifteen years old. She is a beautiful girl [1] bright eyes and long black hair. In some ways we look the same, [2] some students say we are twins. In our class, her math is not good. But she works hard all the time. Now she is doing [3] than before. I hope she can make great progress. I often go to her house. There are many kinds of books and magazines on her bookshelf. She likes reading. Her Chinese is best in our class. She often helps me with Chinese. Liu Mei is an active girl. She's a little more outgoing than me. She likes tennis very much. She is well at tennis. But I'm [4] better than her at ping-pong.

Your score		Your score	
	['and', 'with', 'have', 'has']		['of', 'with', 'sporting', 'having']
	['or', 'so', 'until', 'including']		['because', 'so', 'or', 'but']
	['good', 'well', 'better', 'best']		['worse', 'harder', 'better', 'poorer']
	['zero', 'more', 'little', 'although']		['much', 'more', 'many', 'lot']