

Constructing Open Cloze Tests Using Generation and Discrimination Capabilities of Transformers

Mariano Felice, Shiva Taslimipoor and Paula Buttery

ALTA Institute, Computer Laboratory, University of Cambridge

Cambridge, UK

{mf501, st797, pjb48}@cam.ac.uk

Abstract

This paper presents the first multi-objective transformer model for constructing open cloze tests that exploits generation and discrimination capabilities to improve performance. Our model is further enhanced by tweaking its loss function and applying a post-processing re-ranking algorithm that improves overall test structure. Experiments using automatic and human evaluation show that our approach can achieve up to 82% accuracy according to experts, outperforming previous work and baselines. We also release a collection of high-quality open cloze tests along with sample system output and human annotations that can serve as a future benchmark.

1 Introduction

Open cloze (Taylor, 1953) tests are a common type of exercise where words are removed from a piece of text and must then be filled in by the students without any options to choose from. They are often used in language learning environments as a quick and effective way to test vocabulary, grammar and reading comprehension (Tremblay, 2011; Trace, 2020). However, designing high-quality cloze tests for language learning is a laborious process that involves finding an optimal distribution of gaps based on aspects such as function, distance, number of answers, etc. (ALTE, 2005; 2011).

In this paper, we propose a strategy to construct open cloze exercises using transformer models (Vaswani et al., 2017). Our transformer-based architecture employs two objectives to predict the words that should be gapped in a text passage. Our main objective is standard token classification, where we aim to minimise the error of classifying a token as a gap or not. The second and auxiliary objective is a language-model-based objective whereby we attempt to minimise the language model error when predicting the right answer for each gap. Our solution is based on a pre-trained

ELECTRA (Clark et al., 2020) model that is fine-tuned on the two described objectives in a multi-task scenario.

Our output aims to mimic the style of open cloze tests in the First Certificate in English (FCE) exam¹, which is targeted at learners of English at the B2 proficiency level of the Common European Framework of Reference (CEFR) for languages (Council of Europe, 2001). Unlike other tests, the FCE open cloze task aims to simultaneously test many aspects of grammar and vocabulary that students are expected to know at this level. Since the tests are created from a text passage, they must be skilfully designed in order to ensure an optimal distribution of gaps that adheres to guidelines. A shortened example is shown in Figure 1.

Our system is evaluated under two settings: 1) automatic evaluation, where the generated gaps are compared to gold-standard gaps proposed by test experts, and 2) human evaluation, where the quality of the generated gaps is judged by test experts.

The main contributions of our work are as follows: 1) we are the first to employ transformer models for open cloze test generation, 2) unlike previous studies, we work at the paragraph level, which is a much more challenging task, 3) we propose a multi-task learning approach with two objectives: one is to classify tokens into gaps/non-gaps and the other to minimise the error of re-generating the gapped word, 4) we report state-of-the-art results, outperforming previous work and a strong baseline, 5) we propose additional components to control the structure of the final cloze tests as human experts do, 6) we perform both automatic and human evaluation and 7) we make our test data, system output and human annotations available to the research community².

¹Now known as B2 First: <https://www.cambridgeenglish.org/exams-and-tests/first/>

²Dataset available at <https://github.com/CambridgeALTA/fce-cep-oc>.

Motorbike stunt rider

I work (1) _____ a motorbike stunt rider — that is, I do tricks on my motorbike at shows. The Le Mans race track in France was (2) _____ I first saw some guys doing motorbike stunts. I'd never seen anyone riding a motorbike using just the back wheel before and I was (3) _____ impressed I went straight home and taught (4) _____ to do the same.

Figure 1: Sample FCE open cloze test (shortened).

2 Related Work

While research into automatic cloze test generation is vast (Mostow et al., 2017; Kurdi et al., 2020; Yang et al., 2021), work on open cloze tests for language learning is scarce. Pino et al. (2008) generate open cloze questions using sample sentences from a learners' dictionary based on four linguistic criteria: (grammatical) complexity, well-defined context (collocations), grammaticality and length. A later version of their system adds hints for gapped words (Pino and Eskenazi, 2009). Exercise Maker (Malafeev, 2014) is a rule-based open source system that attempts to emulate exercises in Cambridge English examinations based on the most frequently tested words. Most of the gaps it proposes were found to be useful and the automated exercises were hard to differentiate from authentic tests.

Chinkina et al. (2017) generate open cloze exercises for phrasal verbs by extracting sentences from news articles and generating a pair of questions and answers where the identified particle verbs are gapped. Similarly, Soonklang et al. (2017) gap words in sentences according to their part of speech in order to practise articles, prepositions, etc. Finally, Marrese-Taylor et al. (2018) use LSTMs to build sequence labelling and classification models that decide where to insert a single gap in a single sentence. Automatic evaluation against gold-standard gaps showed the method was effective.

Other work has focused on creating automated cloze tests by controlling aspects of the proposed gaps so that they correlate with a target proficiency level. Lee et al. (2019), for example, manipulate the difficulty of C-tests (open cloze tests with hints, Grotjahn et al. (2002)) by varying the position and word length of the gaps. A similar concept is presented by Settles et al. (2020) and McCarthy et al. (2021), although difficulty is predicted using a machine-learning model that correlates with CEFR levels. In these cases, tests are dynamically adapted to the examinee's proficiency level during the test session. From a different perspective, Felice and Buttery (2019) show that controlling gap entropy

can be useful for designing open cloze tests at different CEFR levels. The work we present in this paper, however, aims to model the more complex task of predicting a full set of gaps at the paragraph level that comply with design and testing principles and is, to the best of our knowledge, the first to employ and adapt transformer-based models for this task.

System evaluation is also challenging, since there is usually more than one potential word in the text that could constitute a good gap. While previous work often made a choice between automatic (Marrese-Taylor et al., 2018) or human evaluation (Malafeev, 2014; Das and Majumder, 2017) for their experiments, we perform both: automatic evaluation to identify the best models during development and human evaluation to measure test quality in the final output.

3 Model

We define open cloze generation as the task of predicting a set of tokens that should be gapped in the text. Unlike previous approaches that work at the sentence level, our models work at the paragraph level (i.e. take the full text as input), since we believe the interactions between gaps can only be optimally captured when the text is processed as a whole rather than sentence by sentence.

Given a text passage, we aim to predict the words that should be gapped in order to create a cloze test that would reliably assess student ability. The task is modelled as a supervised sequence tagging problem where each token is classified as being a good potential gap or not. We employ ELECTRA (Clark et al., 2020), one of the state-of-the-art pre-trained transformer-based language representation models (Wolf et al., 2020). ELECTRA is an extension of BERT (Devlin et al., 2019) with a different pre-training task which is a discriminator (rather than a generator) and aims to detect replaced tokens (rather than generating words for the masks). We believe that this discrimination objective makes it more suitable for our token classification task. Moreover, we also exploit ELECTRA's generation

capabilities as a language model for estimating the answers to the proposed gaps as an auxiliary task. Hence, to make the most of this pre-trained model, we fine-tune it using two training objectives, as depicted in Figure 2:

- ① A *token classification objective* which aims to minimise the error of classifying each token as a potential gap or not.
- ② A *language modelling objective* that aims to minimise the negative log-likelihood of re-generating the words that have been gapped.

The first objective is typical of any standard token classification model and constitutes our key task. In particular, we use ELECTRA’s discriminator head with softmax to tag each word in the input sequence as a ‘good’ gap or not. All the gaps in our training data are replaced with the first intended target answer and labelled positive, while the remaining tokens are labelled negative (A).

The second and auxiliary objective attempts to model our preference for gaps with a restricted number of answers while also ensuring that the original word can be guessed from the context. This is to avoid generating gaps that are too ‘open’ and therefore ineffective, such as a gap that accepts any noun or adjective. Specifically, we mask the words in the positions that are predicted as gaps by the discriminator and use ELECTRA’s generative head to generate the expected words in the blanks (B).

While the input layers are shared between the discriminator and the generator model, the two branches of the system leading to the two objectives are fine-tuned in parallel in a multi-task setting.

4 Extensions

Our neural transformer-based sequence tagging model can be very effective at proposing potentially good gaps, but the task becomes more challenging when we expect the output to meet additional requirements such as no repetitions, no gap interdependence, a minimum distance between gaps and a varied selection of lexico-grammatical items. We address these issues using two complementary strategies: a manipulation of the loss function and a post-processing module.

4.1 Loss manipulation

In order to spread gaps evenly throughout the text, we modify the token-level loss function of our tagging model by imposing a higher penalty on tokens

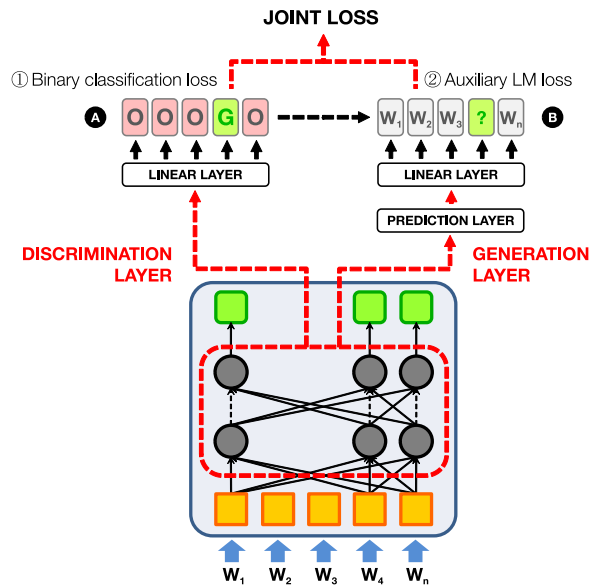


Figure 2: Architecture of our multi-objective ELECTRA-based system. The model is simultaneously trained on two objectives: 1) token classification and 2) LM prediction of gapped words.

that are in close proximity to a gap. Let g be the position of a gap in the sequence, then for each token in position i in the proximity of g , i.e. $|g - i| < D$, the loss function l'_i for the token in position i is defined as:

$$l'_i = l_i * \frac{W}{|g - i|} \quad (1)$$

where W represents the penalty and D is the maximum distance scope for penalisation.³ Equation 1 thus gives more weight to tokens closer to gaps, which results in higher penalisation of their cost functions whenever they are misclassified.

4.2 Post-processing

We also employ a post-processing strategy where we replace the gaps that are repeated in the text with better options. We optimise the choice of these alternative gaps by considering the distance between them and the resulting distribution of gaps with different part-of-speech (PoS) tags.

Our post-processing step can be seen as a re-ranking function. The gap candidates that are originally ranked based on the model’s confidence scores change their ranking to match other desirable requirements of a well-structured cloze test. If the selected n -best gaps include repetitions, our post-processing algorithm randomly chooses one of them at a time and attempts to replace it with

³We empirically set the values of constants D and W to 3 and 3.0 respectively.

a better alternative. An alternative gap is deemed better if 1) its answer is not a repetition of another gapped word, 2) its distance to other selected gaps meets the minimum required distance or is higher than the pairwise distances of the originally selected gaps, and 3) it improves the PoS distribution of the gapped words. The PoS distribution of each new selection of gaps is compared to the average gapped PoS distribution of the cloze tests in the training data using Kullback-Leibler (KL) divergence. A combination of gaps that yields lower KL divergence is assumed to be a better solution.

These extensions to the base model bring our final cloze tests closer to those created by human experts by automatically controlling variables that would otherwise need to be adjusted manually. This makes our solution a fully-automated system that can produce ready-to-use cloze tests from an input text passage.

5 Data

To the best of our knowledge, there are no public datasets of full-text open cloze tests that could be used for our task. The CLOTH dataset (Xie et al., 2018), for example, contains gapped passages designed for language learners, but it is primarily focused on reasoning and reading comprehension and uses multiple choice questions where distractors play a major role, making it substantially different to the task we aim to model.

For this reason, we use a collection of expertly created open cloze tests at the B2 CEFR level that was kindly provided by Cambridge University Press & Assessment (CUP&A) for research purposes. Each task consists of a text passage of no more than 300 tokens, a variable number of gaps (between 8 and 16) and a list of valid answers for each gap (between 1 and 7). During the design process, the tasks undergo extensive quality control and pretesting, so their gaps are guaranteed to be very effective at assessing student ability.

For training, we reconstruct the texts by replacing each gap with its first answer and we split the whole collection into train, dev and test. Details of our dataset are shown in Table 1.

Given the lack of publicly available data, we make our test set available with this paper so as to provide a common benchmark for the task and to encourage further research in this area. All the texts were tokenised and parsed using spaCy v2.3⁴.

⁴<https://spacy.io/>

	Train	Dev	Test
Tasks	356	58	36
Tokens	79,863	12,797	6,621
Gaps	4,565	787	360

Table 1: Number of tasks, tokens and gaps in each section of the data.

6 Experiments

6.1 Setup

We use the pre-trained ELECTRA base discriminator model⁵ with 12 attention heads and 12 hidden layers. Along with all the tokens in the sequences, we also input dependency parsing information to the system. More specifically, we concatenate the ELECTRA representation of each token with the representation of its head in the dependency graph.⁶ On top of the encoding layers, we have two branches that are being learned simultaneously (Figure 2).

The first branch is a simple linear layer that aims to classify each token as a gap or non-gap. For the second branch, we add ELECTRA’s generation layer plus a linear layer which aims to predict the best word from the whole vocabulary as an auxiliary task. We are only interested in predicting the answer words for the gaps. Therefore, we change the input to the second branch by masking the words that are predicted as gaps by the first branch at each step of training. We employ cross-entropy loss on each branch and ignore the loss values for the tokens that are not masked in the second branch. The whole architecture is updated based on the sum of the two losses. Fine-tuning parameters are specified in Appendix B.

6.2 Baselines

We compare our multi-objective ELECTRA model to other systems, namely:

Random baseline Generates a random set of gaps for each task based on the average probability distribution of gapped PoS in the training data.

Exercise Maker Generates gaps using rules and a pre-compiled list of commonly gapped words from a variety of Cambridge English main suite exams (Malafeev, 2014). Set to FCE mode for our experiments.

⁵<https://github.com/huggingface/transformers>.

⁶If the token is the head, then its representation is repeated.

Class	Label	Description
Good	Good	The gap is appropriate, i.e. it is expected to be effective during testing.
Bad	Too close to other gaps	The gap is in close proximity to another gap.
Bad	Too many possible answers	The gap allows too many answers (often more than 5).
Bad	Too many gaps of this type	There are many gaps with the same part-of-speech or testing focus in the text.
Bad	Answers can change meaning	The gap can be filled by answers that would change the meaning of the text, e.g. ‘and’ or ‘but’.
Bad	Answers can have different PoS	The gap can be filled by answers that have a different grammatical function, e.g. ‘which’ or ‘and’.
Bad	Gap depends on another	There is some dependency between this gap and another in the text.
Bad	Repeated gap	There is already another gap testing the same word in the text.
Bad	Phantom gap	The gap does not require an answer for the text to make sense.
Bad	Unacceptable outlier	The gap does not fit in the text for multiple reasons (e.g. inappropriate difficulty).
Bad	Other (please specify)	Any other reason why the gap is considered unsuitable.

Table 2: Labels used in human annotation.

BERT Predicts potentially good gaps using BERT (Devlin et al., 2019) for token classification. We use the pre-trained base model with standard parameters and fine-tune the weights of the whole architecture.

Standard ELECTRA Similar to BERT, it predicts potentially good gaps using a standard pre-trained ELECTRA-base model. This is a single-objective model that is fine-tuned on token classification only.

Both **random** and **Exercise Maker** attempt to generate the same number of gaps per task as defined in the gold standard, although this is not always possible since the required conditions (such as specific words or PoS) are not always met.

6.3 Evaluation

We report precision (P), recall (R) and F_1 scores based on a strict matching between the gaps predicted by our models and those in the gold standard. While this evaluation strategy might seem strict, it has the advantage of being fully automatic, thus avoiding the subjectivity and time required by human evaluation, so we adopt it during development.

In addition to letting the models decide the optimal number of gaps, we also evaluate system performance when we fix the number of predicted gaps for each task to the number of gaps they have in the gold standard. The n -best predicted gaps are chosen based on their confidence scores. In this scenario, P, R and F_1 become the same.

We also report human evaluation by three test experts from CUP&A who volunteered for the task. The experts were asked to label each proposed gap in each task of our test set (a total of 360 gaps) as either good or bad and provide a reason and optional comments for their choice. The list of labels available to our annotators is shown in Table 2.

Model	P	R	F_1
Random baseline	15.29	14.87	15.08
Exercise Maker	23.33	25.79	24.50
BERT	51.16	47.65	49.34
Standard ELECTRA	55.61	46.00	50.35
Multi-objective ELECTRA	57.41	46.25	51.23

Table 3: Models’ performance on the development set.

7 Results and Discussion

7.1 Automatic evaluation

We carry out automatic evaluation by computing P, R and F_1 on our development set. Table 3 reports the results of our multi-objective ELECTRA model (enhanced with dependency information) as well as the random baseline, Exercise Maker, BERT, and the standard single-task ELECTRA. This is our base model, which does not include any loss manipulation or post-processing. In this setting, the number of predicted gaps was decided by each model based on the confidence scores (> 0.5 for the positive class).

Overall, we observe that performance increases with more sophisticated models. Exercise Maker relies on previously seen gaps and so outperforms the random baseline by a large margin. However, it can only create gaps for the 139 words in its predefined FCE word list, missing gaps that are not on that list. Neural transformer-based models are the best, with improvements over Exercise Maker of at least 25 F_1 on our development set. Although the improvement of our multi-objective ELECTRA model over BERT does not seem to be very significant based simply on P, R and F_1 , a closer look at the results reveals that BERT produces a much higher number of repeated gaps (25 compared to 9 by multi-objective ELECTRA) as well as more cases of gaps in close proximity, as shown in Figure 3.

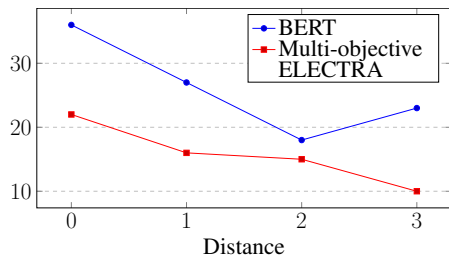


Figure 3: Frequency of pairs of gaps with distance ranging from 0 to 3. Distance is measured by the number of words in between two gapped words. The minimum acceptable number of words between two gaps is 4.

We also perform an ablation study in Table 3 where we compare our multi-objective ELECTRA model to a standard one that does not include our auxiliary language model objective. Results show that the former outperforms the latter on all metrics, confirming that the addition of the LM objective is clearly beneficial.

Table 4 shows the performance of our multi-objective ELECTRA model as we increase the n -best list of gaps according to their confidence score. The first row indicates the results of the system when it is forced to predict the exact same number of gaps per task as in the gold standard.⁷ This causes P and R to be the same. As we expect, the results show that the number of gaps in the gold data is actually the optimal number to achieve the best F₁ score.

Although our multi-objective model shows good performance based on automatic evaluation, a closer look at the output reveals that the structure of the cloze tests is far from ideal as they often contain repetitions and gaps that are too close to each other, aspects that are carefully controlled in the gold standard. Table 5 shows that system performance effectively improves as we add the extensions proposed in Section 4, indicating that global aspects of the task are not properly captured by our initial model and require further manipulation.

In order to make the structure of our output as similar as possible to our target tasks, we fix the number of predicted gaps for each task to the number of gaps they have in the gold standard. Note that P and R are the same in this setting so we only report F₁. The effect of this decision is shown in Table 6. We can see that adding loss manipulation to our model decreases the number of adjacent gaps from 40 to 23, but increases the number of repeated

⁷The number of gaps can vary per passage (see Appendix A).

# of predicted gaps	P	R	F ₁
As-in-gold	54.26	54.26	54.26
10	56.72	42.80	48.14
15	51.49	56.93	54.07
20	44.83	66.07	53.42
30	35.63	78.78	49.07

Table 4: Results of multi-objective ELECTRA when we predefine the number of predicted gaps.

	P	R	F ₁
Multi-objective ELECTRA	57.41	46.25	51.23
+ loss manipulation	47.87	59.85	53.19
+ post-processing	48.42	60.23	53.68

Table 5: Effect of loss manipulation and post-processing on our multi-objective ELECTRA model.

gapped words from 18 to 33. The decline in the restricted F₁ based on automatic evaluation is not favourable, but we make this sacrifice at the price of achieving a better-structured final test.

After adding post-processing for repeated gaps, we observe that, although overall F₁ performance drops slightly, the number of repeated gapped words decreases favourably from 33 to 9 (Table 6). It also creates a better spread of gaps, as shown by a lower KL-divergence between the average PoS distribution of the output and that of the gold standard (0.55 with post-processing as opposed to 0.59 without it). Post-processing also removes two cases in the development set where the gaps do not meet the minimum 4-word distance.

It is worth recalling that these extensions are highly effective when we do not restrict the number of predicted gaps. Table 5 shows that they significantly improve R, which results in higher overall F₁.

As a result of these experiments, we stick with our post-processing approach for the rest of our experiments and use it to produce the output submitted for human annotation.

7.2 Human evaluation

Following our intuition that test experts could find more value in our system than initially shown by our automatic evaluation, we asked a panel of three test experts to judge the quality of the gaps produced by our extended model on the test set. Inter-annotator agreement on gap classification (good/bad) was found to be moderate (percent agreement is 75.93%, Randolph’s free-marginal kappa is 0.52 (Randolph, 2005)).

	Restricted F_1	Repeated gaps	Adjacent gaps
Multi-objective ELECTRA	54.26	18	40
+ loss manipulation	51.59	33	23
+ post-processing	51.33	9	23

Table 6: Analysis of our model after adding extensions: loss manipulation and post-processing.

Auto	Ann. 1	Ann. 2	Ann. 3
53.89	82.50	75.83	77.50

Table 7: Accuracy of our extended multi-objective ELECTRA model based on automatic and human evaluation on the test set.

Unlike in automatic evaluation, we only report accuracy for our human experiment. System performance using automatic and human evaluation is compared in Table 7 (reported individually for each annotator). These results show that performance increases dramatically when the output is judged by human experts, confirming our suspicion that performance is underestimated by automatic evaluation and that there are many other words in the texts that could constitute equally useful gaps apart from those in the gold standard. With system accuracy ranging between 75% – 82% for human judgements, we can conclude that at least 7 out of 10 gaps proposed by our system are considered good by our experts.

We observed that differences between annotators’ judgements and the gold-standard can occur for many reasons, e.g.:

- non-gaps in the gold standard are not necessarily bad gaps,
- gold standard gaps are derived from pilot testing while annotators’ gaps are derived from their expertise,
- previous judgements by the annotators can affect the judgement of new gaps (e.g. choosing the best of two close gaps), etc.

Annotator accuracy against the gold standard ranges between 50% – 60%.

Following our classification in Table 2, we analysed the reasons why some gaps were not considered good by the annotators. Figure 4 shows the average frequency of the different reasons given by the annotators for rejecting a gap proposed by our system. Examples are included in Appendix C.

The most frequent reason is the violation of the minimum required distance between two gaps (42.43%). Although our loss-manipulation ap-

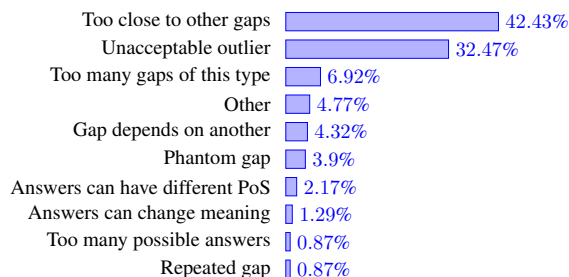


Figure 4: Average frequency of the reasons given by the annotators for rejecting a gap.

proach was successful in reducing these cases, we did not attempt to eradicate them completely since there are many factors at play when choosing more appropriate gaps than just distance. In many cases, gaps in close proximity test different words in the same phrase (e.g. *take part in, in addition to*, etc.) so we preferred to keep these cases and encourage annotators to comment on their preferences. Reiterations, on the contrary, are much better handled, accounting for only 0.87% of all bad gaps.

The second most frequent reason is ‘unacceptable outlier’ (32.47%), which normally accounts for cases where the difficulty of the gap is considered inappropriate for the target proficiency level (B2 in this case). This is an interesting phenomenon, since the fact that the text as a whole pertains to a given CEFR level does not guarantee that the gaps created will always be appropriate for the level. The remaining reasons are substantially less frequent than the first two and mostly related to aspects that were not explicitly controlled in our models, except for the third topmost reason (‘Too many gaps of this type’) which we did control by comparing PoS distributions. These results show that our system is able to capture many aspects of the task that were not explicitly modelled.

Finally, we compared system accuracy per task computed from annotators’ judgements vs. the gold standard. Average correlation across all annotators was found to be very weak (Pearson’s $r = 0.0558$, Spearman’s $\rho = 0.1474$). This suggests that automatic scores are not a good proxy for human perception, with experts being much more positive about our model’s output (as shown in Table 7).

7.3 Predictions by Gapped Word Frequency

We found that our model does not overfit to words that are most frequently gapped in the training data, with correlation between gapped word frequency

Gardening

It is early summer , the season of abundance , when my garden is at its fullest . Flowers are in bloom and the grass is growing so fast that half an hour after cutting it , I seem to be back where I started . This year for the first time I am attempting to grow my own vegetables , an attempt that has so far proved very successful . My vegetable plants have been yielding an abundance of produce , in fact much more than I can possibly consume myself . I ’m convinced that you cannot plant even a single tomato without feeling a connection to the earth and to the countless generations who have worked the land before you . To plant seeds and then to harvest what you have grown gives a deep sense of satisfaction . I believe that many doctors and mental health organisations all around the world now recognise the value of gardening to the well-being of those who take part in this activity .

Figure 5: Sample output of our extended ELECTRA model. Darker shades of red indicate higher confidence in inserting a gap. Predicted gaps are framed in black while gold standard gaps are in yellow font.

PoS	Proportion in TEST	P	R	F ₁
ADP	20.59%	50.00	43.24	46.38
ADV	14.17%	57.69	58.82	58.25
DET	13.89%	56.41	44.00	49.44
SCONJ	13.89%	59.09	78.00	67.24
AUX	10.83%	45.83	28.21	34.92
PRON	9.44%	47.92	67.65	56.10
ADJ	4.44%	60.00	75.00	66.67
NOUN	3.33%	77.78	58.33	66.67
NUM	2.78%	61.54	80.00	69.57
CCONJ	2.50%	55.56	55.56	55.56
VERB	2.22%	50.00	50.00	50.00
PART	1.67%	0.00	0.00	0.00
INTJ	0.28%	50.00	100.00	66.67

Table 8: Performance by PoS on the test set based on automatic evaluation.

and F₁ scores in the test set being negligible (Pearson’s $r = 0.0108$, Spearman’s $\rho = 0.0915$).

Interestingly, while our model was unable to predict gaps not previously seen in the training data (*turned*, *amount*, *pushed* and *started*), it did predict a (previously unseen) gap for the word *fewer*, which did not match the gold standard but was unanimously deemed good by our annotators.

7.4 Predictions by PoS

We also classified predictions based on their PoS tags⁸ and report performance in Table 8. The most frequently gapped PoS tags in our datasets correspond to closed word classes (such as ADP, DET, SCONJ, AUX, etc.), which is expected given that our open cloze tests are mostly focused on testing grammar rather than vocabulary. The best predicted classes, however, are NUM, SCONJ, NOUN, ADJ and INTJ which on closer inspection turn out to

⁸Using the Universal Dependencies tagset: <https://universaldependencies.org/u/pos/>

be very restricted classes: NUM includes only the word *one*, INTJ only the word *like*, SCONJ only a few subordinating conjunctions while NOUN and ADJ, despite being open classes, are limited to words used in common constructions such as *order* (*in order to*) or *same* (*the same*).

The two worst performing classes are PART (the particles *to* and *not*) and AUX (auxiliary verbs) and, once again, we conjecture that these words are so common in the language and in non-gapped positions that the model is unable to get them right most of the time. The remaining PoS classes vary in performance but we found only very weak correlation between PoS gap frequency in the test set and F₁ scores (Pearson’s $r = 0.1932$, Spearman’s $\rho = 0.1350$).

When we look at human annotations on the test set, however, performance by PoS is consistently higher and more even across the board. If we require that gaps are rated ‘good’ by at least two annotators, accuracy values range between 75% and 100% for all PoS, with a mean of 85%.

Under these conditions, the best performing classes are NOUN (100%), INTJ (100%) and ADJ (95%), which agree with automatic evaluation. Out of these, only NOUN achieves perfect accuracy across all annotators. The worst performing classes are PRON (77%), NUM (77%) and VERB (75%) as opposed to the previous AUX and PART counterparts (now 79% and 83% respectively). When we require agreement by all annotators, the worst overall class is CCONJ with 44%.

7.5 Qualitative Analysis

Figure 5 shows the output of our model for a sample text passage, where darker red indicates higher confidence in inserting a gap. The final model’s

predictions have a black frame (*at, in, so, after, etc.*) while the gold standard gaps are in yellow font (*at, in, so, etc.*). There are 8 matched gaps out of 11 in this example, yielding 72.73% accuracy.

As can be seen in the figure, our model is able to identify appropriate gap candidates, even if they do not match the gold standard. In fact, annotators considered all the unmatched gaps in this example (*after, for* and *take*) to be good and the second matched gap (*in*) to be inappropriate. It is also interesting to see how the model prioritises function words and content words that are highly restricted in context (such as *take* or *part*), skilfully avoiding general gaps that could accept multiple answers and would be less effective for testing purposes.

8 Conclusion and Future Work

We described the first transformer-based approach to open cloze test generation. Our ELECTRA-based model is trained on two objectives: token classification (gap/non-gap) and language modelling (for predicting the expected answer). The model is further improved by manipulating the loss function and post-processing the results.

System accuracy using automatic evaluation is 53.89% while human evaluation ranges between 75% – 82%, showing that at least 7 out of 10 gaps predicted are considered useful by experts. A detailed analysis of results reveals a few structural problems such as gaps in close proximity and inappropriate difficulty, which we plan to address in future work. Our test data and human annotations are released with this paper.

Acknowledgements

The authors are immensely grateful to Louise Gilbert, Sally Moore and Clare Williams from CUP&A for their annotations. This paper reports on research supported by Cambridge University Press & Assessment, University of Cambridge.

References

Maria Chinkina, Simón Ruiz, and Detmar Meurers. 2017. [Automatically generating questions to support the acquisition of particle verbs: Evaluating via crowdsourcing](#). In *CALL in a climate of change: adapting to turbulent global conditions – short papers from EUROCALL 2017*, pages 73–78.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training Text Encoders as Discriminators Rather](#)

[Than Generators](#). In *International Conference on Learning Representations*.

- Council of Europe. 2001. *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press, Cambridge.
- Bidyut Das and Mukta Majumder. 2017. Factual open cloze question generation for assessment of learner’s knowledge. *International Journal of Educational Technology in Higher Education*, 14:1–12.
- 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.
- Mariano Felice and Paula Buttery. 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, Varna, Bulgaria. INCOMA Ltd.
- Rüdiger Grotjahn, Christine Klein-Braley, and Ulrich Raatz. 2002. C-tests: an overview. In James A Coleman, Rüdiger Grotjahn, and Ulrich Raatz, editors, *University language learning and the C-Test*, pages 93–114. AKS-Verlag, Bochum, Germany.
- Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. [A systematic review of automatic question generation for educational purposes](#). *I. J. Artificial Intelligence in Education*, 30(1):121–204.
- Ji-Ung Lee, Erik Schwan, and Christian M. Meyer. 2019. [Manipulating the difficulty of C-tests](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 360–370, Florence, Italy. Association for Computational Linguistics.
- Alexey Malafeev. 2014. [Language exercise generation: Emulating cambridge open cloze](#). *Int. J. Concept. Struct. Smart Appl.*, 2(2):20–35.
- Edison Marrese-Taylor, Ai Nakajima, Yutaka Matsuo, and Ono Yuichi. 2018. [Learning to automatically generate fill-in-the-blank quizzes](#). In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 152–156, Melbourne, Australia. Association for Computational Linguistics.
- Arya D. McCarthy, Kevin P. Yancey, Geoff T. LaFlair, Jesse Egbert, Manqian Liao, and Burr Settles. 2021. [Jump-starting item parameters for adaptive language tests](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 883–899, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Jack Mostow, Yi-Ting Huang, Hyeju Jang, Anders Weinstein, Joe Valeri, and Donna Gates. 2017. [Developing, evaluating, and refining an automatic generator of diagnostic multiple choice cloze questions to assess children’s comprehension while reading](#). *Natural Language Engineering*, 23(2):245–294.
- Association of Language Testers in Europe (ALTE). 2005. [Materials for the guidance of test item writers](#). Technical report, Association of Language Testers in Europe.
- Association of Language Testers in Europe (ALTE). 2011. [Manual for language test development and examining](#). Technical report, Association of Language Testers in Europe.
- Juan Pino and Maxine Eskenazi. 2009. [Measuring hint level in open cloze questions](#). In *Proceedings of the Twenty-Second International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, Sanibel Island, Florida, USA. AAAI Press.
- Juan Pino, Michael Heilman, and Maxine Eskenazi. 2008. [A selection strategy to improve cloze question quality](#). *Intelligent Tutoring Systems for Ill-Defined Domains: Assessment and Feedback in Ill-Defined Domains*, page 22.
- Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss’ fixed-marginal multirater kappa. *Online submission*.
- Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. [Machine learning–driven language assessment](#). *Transactions of the Association for Computational Linguistics*, 8:247–263.
- Tasanawan Soonklang, Sunee Pongpinigpinyo, Weenawadee Muangon, and Sirak Kaewjamnong. 2017. [Automatic question generation system for english exercise for secondary students](#). In *Proceedings of the 25th International Conference on Computers in Education (ICCE 2017)*, pages 890–895, New Zealand. Asia-Pacific Society for Computers in Education.
- Wilson L Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.
- Jonathan Trace. 2020. [Clozing the gap: How far do cloze items measure?](#) *Language Testing*, 37(2):235–253.
- Annie Tremblay. 2011. Proficiency assessment standards in second language acquisition research: “clozing” the gap. *Studies in Second Language Acquisition*, 33(3):339–372.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS) 30*, pages 5998–6008. Curran Associates, Inc.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 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, Brussels, Belgium. Association for Computational Linguistics.
- Albert C. M. Yang, Irene Y. L. Chen, Brendan Flanagan, and Hiroaki Ogata. 2021. [Automatic generation of cloze items for repeated testing to improve reading comprehension](#). *Educational Technology & Society*, 24(3):147–158.

A Dataset composition

# Gaps	Train	Dev	Test
8	4	0	0
9	39	2	9
10	71	2	18
11	38	12	9
12	4	6	0
13	58	9	0
14	10	2	0
15	0	0	0
16	132	25	0
Total	356	58	36

Table A.1: Distribution of the number of gaps per task in each section of the data.

# Answers	Train	Dev	Test
1	3637	639	296
2	689	111	45
3	147	23	16
4	77	11	2
5	11	3	1
6	3	0	0
7	1	0	0
Total	4565	787	360

Table A.2: Distribution of the number of answers per gap in each section of the data.

B Model parameters

Parameters	BERT	Multi-objective ELECTRA
Learning rate	3×10^{-5}	3×10^{-5}
Batch size	1	1
Number of epochs	4	4
Training steps	$\frac{n}{b} \times e$	$\frac{n}{b} \times e$

Table B.1: Model parameters used for the experiments. n : the number of training examples; b : batch size; e : number of epochs.

C Human labelling examples

Reason for rejection	Example gap in context	Annotator comments
Too close to other gaps	... the thousands of questions I asked <u>as a child were</u> met not by impatient answers ...	Minimum distance is not met.
Too many possible answers	... and <u>does</u> not sound threatening.	Many verbs could fit in this gap: <i>does, may, might, should, will</i> , etc.
Too many gaps of this type	... the country <u>where</u> the largest number of bamboo varieties grow naturally...	Too many relative pronouns are tested in the task.
Answers can change meaning	... the petals were narrower and <u>less</u> clearly separated ...	The word <i>more</i> also fits.
Answers can have different PoS	The Indian bansuri bamboo flute, <u>when</u> played by a master musician, ...	Other possible answers are <i>often, usually, normally</i> , etc.
Gap depends on another	<u>What</u> I love most about being on a horse <u>is</u> that ...	The second gap depends on the first.
Repeated gap	..., <u>although</u> she later became a biologist.	The task has another gap where <i>although</i> is a possible answer.
Phantom gap	The name actually refers to the statuette <u>which</u> all of the winners receive.	<i>Which</i> can be omitted.
Unacceptable outlier	The school was by <u>no</u> means an overnight success; ...	The phrase <i>by no means</i> is at the C1 CEFR level.
Other (please specify)	It is sometimes said that animals use language.	Avoid having a gap for the very first word in the text.

Table C.1: Example of the different reasons given by the annotators for rejecting a gap proposed by our system.