1. Sebastian JĘDRZEJEWSKI, 2. Robert SZMURŁO

Warsaw University of Technology, Institute of Theory of Electrical Engineering and Information Systems
ORCID: 2. 0000-0001-8041-4438

doi:10.15199/48.2025.03.57

Cloze-tests generation for foreign language learning using transformer networks

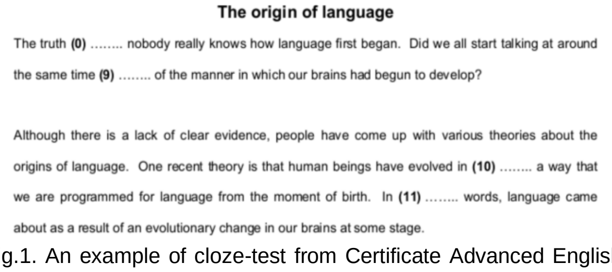
***Abstract**. The aim of the paper is to evaluate the transformers network LLM for automatic generation of tests for English language learners with classical approaches. The kind of test we investigate is referred to as cloze test, which are paragraphs of text with gaps which should be filled in by the learners. In the paper, we compare recurrent neural networks to transformer networks (BERT and ELECTRA). Additionally the authors make the training and testing datasets available publicly. The approach related to application of LLMs is based on paper by Felice at al. [2]. In the presented research we extend the loss function and apply extra metrics based on Kullback-Leibler Divergence Loss to improve space distribution of the gaps.*

***Streszczenie.** Celem artykułu jest zbadanie możliwości sieci transformerowych LLM do automatycznego generowania testów dla osób uczących się języka angielskiego oraz porównanie wyników z metodami klasycznymi. W artykule badany jest rodzaj testu, który składa się z akapitów tekstu z lukami, które powinni wypełnić uczący się. W artykule porównane są wyniki uzyskiwane za pomocą sieci rekurencyjnych neuronowych oraz sieci transformerowych (BERT i ELECTRA). Podejście związane z zastosowaniem sieci LLM opiera się na artykule Felice i in. [2]. W przedstawionych badaniach autorzy rozszerzają funkcję straty o dodatkowe metryki oparte na stracie dywergencji Kullbacka-Leiblera w celu poprawy rozkładu przestrzennego luk. (**Generowanie testów Cloze'a do nauki języków obcych przy użyciu sieci Transformer**)*

Keywords: cloze-test, recurrent neural networks, transformer networks, LLM.
Słowa kluczowe: test na uzupełnianie luk w tekście, rekurencyjne sieci neuronowe, sieci transformerowe, LLM.

Introduction

The cloze-test is a popular tool for testing and training for foreign language learners. An example of such a test is presented in Figure 1. The method was invented by Taylor [1] in the early 1950s. The term 'cloze' stems from the Gestalt psychology theory of the principle of closure, which mentions the human tendency to perceive complete patterns from partially hidden or incomplete patterns [9]. The paper focuses on the English language because of training datasets availability, but general methodology is universal and could be applied to any language. The cloze test is a kind of test in which learners have paragraphs of text with gaps and the task is to find the correct missing word (fill-in-the-blanks). Cloze tests are mostly popular among language learners due to its value for grammar and vocabulary knowledge verification. Cloze-tests are massively used for evaluation of text input understanding of various NLP models [7]. The paper focuses however only on real examinations and learning tests. In such case, the generation of cloze tests is mostly done manually by teachers which is a time-consuming task to produce significant number of tests. There are many tools, both online and offline which allow uploading a text and clicking on a word or expression to indicate which word a teacher identifies as a candidate for a gap. The goal of the paper is to investigate the methodological quality of cloze tests generated automatically with Machine Learning approaches: classical recurrent neural networks and transformer models (BERT and ELECTRA).


The origin of language

The truth **(0)** …….. nobody really knows how language first began. Did we all start talking at around the same time **(9)** …….. of the manner in which our brains had begun to develop?

Although there is a lack of clear evidence, people have come up with various theories about the origins of language. One recent theory is that human beings have evolved in **(10)** …….. a way that we are programmed for language from the moment of birth. In **(11)** …….. words, language came about as a result of an evolutionary change in our brains at some stage.

Fig.1. An example of cloze-test from Certificate Advanced English (CAE)

The difficulty behind the automatic cloze-test generation is to generate a test satisfying a methodological quality. Thus, the random word choice is unacceptable. When using the supervised learning one could use only an accuracy or F1 score. But relying solely on the F1 score when assessing the quality of a test is not the best idea. This is illustrated by the example of predicting gaps for a short text by 2 imaginary models. The predictions of the first model are shown in Figure 2. The correct gaps are marked in yellow, the model choices are marked in purple, and the correct choices are marked in purple and yellow. The F1 score for this case is 66.7%, so it is not a very bad result. However, assessing the quality of this test deeper, one can conclude that it leaves something to be desired. Firstly, two gaps occur next to each other, which will cause difficulties in inserting the appropriate consecutive words when solving. Secondly, the word '*of*' was classified as a gap twice, which is also an imperfection of this test. Figure 3 presents the results of a second model. The coloring remains the same. F1 can be calculated again, and this time the result is 44.4%. This is significantly lower than in the case of the first model. This time, however, the distribution of gaps is better (there are no gaps too close to each other), and each gap is unique. The model 'incorrectly' classified the words *'be', 'instead'* and *'at'* as gaps, which in the author's opinion is not a bad choice at all and the test presented in Figure 3 is probably a better quality test than the ground truth one, despite the worse F1 result. Thus, in the evaluation section the authors analyzed all three metrics: F1 score, gaps distribution and words repetitions separately.

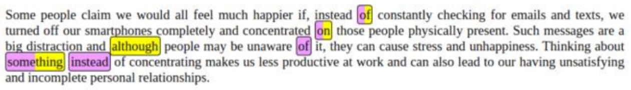

Some people claim we would all feel much happier if, instead of constantly checking for emails and texts, we turned off our smartphones completely and concentrated on those people physically present. Such messages are a big distraction and although people may be unaware of it, they can cause stress and unhappiness. Thinking about something instead of concentrating makes us less productive at work and can also lead to our having unsatisfying and incomplete personal relationships.

Fig.2. Predictions of gaps (purple) and 'ground truth' (yellow) for imaginary model 1

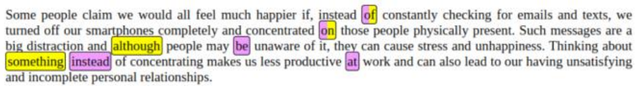

Some people claim we would all feel much happier if, instead of constantly checking for emails and texts, we turned off our smartphones completely and concentrated on those people physically present. Such messages are a big distraction and although people may be unaware of it, they can cause stress and unhappiness. Thinking about something instead of concentrating makes us less productive at work and can also lead to our having unsatisfying and incomplete personal relationships.

Fig.3. Predictions of gaps (purple) and 'ground truth' (yellow) for imaginary model 2

Related work

The problem of automatic English test generation dates back to 1980s and 1990s [8]. There have been many approaches to automatic fill-in-the-blanks type test generation like for example rule based "nth-word deletion" in a text with words having assigned class tags. Where the tags were coming from Automatic Grammatical Tagging System or others [8]. The recent advances in Machine Learning and emergence of deep recurrent networks and transformer models made possible creation of such tests based just on training data without need of engineering rules. In [9] the authors apply a masked language AI model and the "Gini coefficient" and develop an algorithm named CLOZER. The model focuses mostly on answer uniqueness. Mariano Felice et al., in [2] proposes the first multi-objective transformer model for constructing open-cloze. The transformer based architecture employs two main objectives: standard token classification, where the model aims to minimize the error of classifying a token as gap or non-gap and to minimize the error when predicting the right answer for each gap. The aim of the authors was to mimic the style of open cloze tests in the First Certificate in English exam.

The former paper by Mariano Felice et al. is a foundation for the presented paper. The main contributions of our work are as follows:

1) compare and evaluate the multi-objective transformer model with recurrent neural networks,
2) evaluate Kullback-Leibler Divergence Loss to improve gapping,
3) a new, open training dataset created from online available tests for Cambridge certificates.

Models

We can think of generating open cloze tests as a standard token classification task, similar to parts of speech recognition. Instead of parts of speech, we have two labels: a word that should be predicted as a gap or a word that shouldn't be. Therefore, we have implemented three types of models trained for the binary classification task: one classic RNN and two transformer models (BERT and ELECTRA). Each language test has been tokenized using either a tokenizer from TensorFlow[1] (for the RNN model) or a built-in tokenizer specified for transformer models from Hugging Face Transformers[2] (for BERT and ELECTRA).

a. RNN

This is a simple recurrent neural network with LSTM cells and a bidirectional structure. The number of input neurons is set to the maximum number of tokens in a single input text from the training and validation tests. We determined experimentally that one hidden layer with 128 neurons is optimal for our dataset. The output layer is a dense layer that classifies a token as a gap word or a non-gap word.

b. BERT

A standard pretrained BERT model (introduced in the paper by Devlin [3]) that has been fine-tuned for the token classification task using our training dataset.

c. ELECTRA

A standard ELECTRA model was introduced in the paper by Clark [4]. It is an extension of the BERT model but with a different training strategy. In our research, we have implemented a multi-objective ELECTRA model proposed by the authors of [2]. The assumption of this approach is to train the model for two tasks simultaneously:

1) a token classification task, similar to the two previous models;
2) prediction of suitable words for gaps generated by the first objective.

The purpose of the latter is to ensure that the words for generated gaps are possible to guess, avoiding gaps where one can insert any noun or adjective. The first part of the model (the discriminator model) generates a test with some gaps in it. Then, a special token is inserted in the gap positions, used by the generator model to mark which words it must predict. True labels for the second model are the words that were removed from the original test by the discriminator. For both parts, cross-entropy loss is calculated between their predictions and true labels. The model is trained simultaneously for these two objectives, and all weights are updated based on the joint loss. This is the approach taken by the authors of [2], and we, inspired by this idea, decided to explore its capabilities and compare it with different approaches.

Extensions

Generating a high-quality open cloze test is not only about predicting suitable words for gaps but also about appropriately distributing the gaps and ensuring a variety of words chosen to be gaps. We tackled these problems similarly to the authors of [2] by applying a loss manipulation extension and a post-processing phase for the multi-objective ELECTRA model. Moreover, we extended the solutions described by the authors by using Kullback-Leibler divergence instead of loss manipulation to address the first issue.

a. Kullback-Leibler Divergence Loss

The alternative extension to loss manipulation proposed by us uses the Kullback-Leibler (KL) divergence loss. The KL divergence is a measure that shows the difference between the probability distribution P and the reference (or model) distribution Q. Its general formula is:

$$D_{\mathrm{KL}}(P \square Q) = \sum_{x} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{1}$$

More information about this divergence can be found in the paper by Jonathon Shlens et. al. [5]. For our model, the base distribution P is the average gap distances distribution in our training dataset. It includes the probability of occurring two gaps with X words between them, and for small X (e.g., 0, 1), it has smaller values as it is very unlikely to find two gaps with only 1 or 2 words between them. Such a distribution is calculated during training for model predictions, and the KL divergence loss represents the difference between the model gaps distribution and the base one. Eventually, this loss is added to the discriminator and generator losses.

b. Post-processing

Similar to the approach described by the authors in [2], we introduced a post-processing phase to enhance the quality of the generated tests. This phase operates on model predictions and does not involve further model training. It has two primary objectives: to generate a test

[1] https://www.tensorflow.org/api_docs/python/tf/

[2] https://huggingface.co/

Table 3. Different dap distribution metrics

Model	Predicted gaps	Adjacent gaps	Too close gaps	Too close gaps percentage	Ideal gap positions std
RNN	204	14	24	11.76	19.41
BERT	228	17	46	20.18	22.70
ELECTRA	168	8	18	10.71	25.91
ELECTRA with loss manipulation	144	2	11	**7.64**	24.51
ELECTRA with KL divergence	196	15	29	14.80	22.76
ELECTRA with post-processing	172	1	**10**	**5.81**	-

with N gaps (if feasible) and to minimize the number of repeated gaps.

During this phase, we adjust the threshold for treating a word as a gap from 50% to 40%, as the models sometimes predict too few gaps for a given text.

Initially, two groups of gaps are created. The first group is considered the best set of gaps for generating a high-quality test, while the second group consists of alternative words that might be moved to the first group under certain conditions.

Initially, the first group of gaps is composed of the model's predictions as if without post-processing, and the alternatives are words that the model classified as gaps with confidence between 40% and 50%. The alternatives are always kept sorted in descending order of confidence so that the most probable gaps are chosen first. Next, words are added to or removed from the first group to ensure it contains N words (if there are sufficient words in total). An alternative can be moved to the first group only if it is not a repetition of another word there and if its distance from the already predicted gaps is at least four words.

Lastly, the predicted gaps are checked in random order. If any word is found to be a repetition of another gap, the alternatives are scanned to find a suitable replacement. If an alternative meets the conditions mentioned above, it is swapped with the predicted gap from the first group. At the end of this process, the first group is returned as the result of the post-processing phase.

Table 1. Dataset tests and gaps distribution

	Training	Validation	Test
Tests	264	43	27
Gaps	2281	388	230

Data

An important part of our research is to gather useful and high-quality examples of open cloze tests to train our models. There are no public datasets for such tasks, so we propose our own dataset, which mainly consists of freely available tests created for English learners who prepare for Cambridge certificates.

Tests acquired from different sources have been converted from a user-friendly format to JSON, where places in the text where gaps should be located have been marked with a special character. The text passages in the dataset contain several gaps (at least 8) and a list of possible answers for each gap. During training, we use only the first valid answer from the list, marking it as a potential gap, and the other words in the texts are treated as usual words. We split our tests into train, validation, and test sets. The distribution and contents of the dataset has been summed up in table 1.

Researchers from [4] were provided by Cambridge University Press & Assessment (CUP&A) with open cloze tests prepared by experts therefore their training set is proprietary, and they are not allowed to share it. However, we go a step further and make our training and test datasets publicly available since they consist of tests accessible to everyone.

Experiments

For each of our built models (including ELECTRA with extensions), we apply various metrics to evaluate their usefulness and to compare them against each other.

a. Automatic Evaluation

In this step, we calculate standard precision (P), recall (R), and F1 scores for validation and test sets, as is typical for any machine learning task. The model is provided with a passage and determines which words in the text should be marked as gaps, thereby selecting the number of gaps autonomously. Since we have only two possible labels for each token, standard binary metrics are computed between model predictions and true labels from the dataset.

While this approach is effective for any token classification task, our task is somewhat unique. For example, in part-of-speech recognition, if a model classifies a word as an adjective instead of a noun, it is clearly incorrect. However, in our task, if the model predicts a word as a gap that should not be according to our dataset, it does not necessarily mean the model is flawed, provided the gaps are not too close to each other and there are no repetitions. Open cloze tests are often created by experts, and it is likely that two experts given the same passage would choose different sets of potential gaps without either being wrong. These tests must meet conditions such as appropriate gap distribution and lack of repetitions. Consequently, our F1 scores are not high, so we apply additional metrics to further evaluate our models' performance.

b. Gap Distribution

For gap distribution, we check the number of gap pairs that are too close to each other. Gaps are predicted by the model for every text in the test dataset. We define the distance between gaps as the number of words between them, with an acceptable distance being at least four words. We consider the total number of predicted gaps for the test dataset, the number of adjacent gaps (with a distance of zero), the number of gaps that are too close (with a distance of less than four), and the percentage of the latter relative to all gaps. Additionally, we calculate the standard deviation between the positions of the gaps predicted by the model and the ideal gap positions. Although the distances may be acceptable, gaps could be clustered in one part of the text instead of being evenly distributed. Ideal gap positions take text length and the number of gaps into consideration, assuming equal distances between gaps. While a zero value for this metric is not possible, small or very high values can help draw conclusions about gap distribution.

c. Gap Repetition

This metric is straightforward. Gaps are predicted for texts in the test dataset, and the number of repeated gap pairs is counted. We consider the total number of predicted gaps, the number of repeated gaps, and the percentage of repeated gaps.

Results

a. Automatic evaluation

Each transformer model was trained using learning rates ranging from 2e-5 to 6e-5. For each learning rate, the training process was repeated three times because the dataset is relatively small, and fluctuations in F1 scores were observed. The average F1 score was calculated for each learning rate, and the best learning rate was determined as the one with the highest average F1 score. Table 2 shows the results of the model trainings. In general, the transformer models performed similarly, and they consistently outperformed the neural network. Our modification of the loss function turned out to perform the best, although the differences are not so significant. We assume that F1 in our task is useful for tuning hyperparameters and assessing whether the model avoids choosing random words for gaps, but it is not the primary metric. To evaluate the quality of the generated tests, we also consider other measures.

Table 2. Automatic evaluation scores

Model	P	R	F1
RNN	31.94	29.64	30.75
BERT	40.04	36.49	38.12
ELECTRA	42.87	32.90	37.22
ELECTRA with loss manipulation	46.32	29.74	36.22
ELECTRA with KL divergence	45.03	34.20	**38.81**
ELECTRA with post-processing	45.83	28.45	35.11

b. Gap distribution

Table 3 presents the metrics results for gap distribution as discussed earlier. Importantly, the addition of the loss manipulation extension significantly reduced the number of gaps with a distance of less than four words, making it the most effective model in this aspect. Furthermore, incorporating a post-processing phase further enhances gap distribution, as alternatives are selected only if they maintain an appropriate distance from other gaps.

However, despite achieving the highest F1 score, the model with KL divergence was found to generate even more gaps in close proximity compared to the standard model. Nevertheless, it effectively disperses gaps more evenly throughout the text, as indicated by the standard deviation of ideal gap positions[3].

It is noteworthy that BERT tends to generate a lot of gaps near each other, which shows that even though it performed better in terms of F1 score than ELECTRA, it does not necessarily guarantee to generate high-quality tests. Even RNN did not generate so many gaps close to each other.

c. Gap repetition

The occurrence of repeated gaps was assessed on the test set, and the results are detailed in Table 4. Once more, ELECTRA with loss manipulation generates the tests with the smallest number of repeated gaps and adding post-processing phase showed minimal impact on this measure. Standard BERT again performed poorer than multi-objective ELECTRA but this time the difference is not so radical as in the case of gaps distribution.

Table 4. Repeated gaps metrics

Model	Predicted gaps	Repeated gaps	Repeated gaps percentage
RNN	204	15	7.35
BERT	228	11	4.82
ELECTRA	166	6	4.38
ELECTRA with loss manipulation	137	2	**1.46**
ELECTRA with KL divergence	190	6	3.16
ELECTRA with post-processing	172	3	1.74

Summary

In the paper a method of generating cloze-tests using classical deep recurrent neural networks and modern transformer based architecture methods were compared and analyzed. The authors presented a novel approach to improve gap distribution using Kullback-Leibler as an additional term in the loss function. The results have shown that it improves the F1 score (see table III). The presented results confirmed the ability of transformer networks to produce high quality test from the methodological point of view.

***Authors**: Sebastian Jędrzejewski, dr inż. Robert Szmurło, Politechnika Warszawska, Wydział Elektryczny, ul. Koszykowa 75, 00-661 Warszawa, E-mail: robert.szmurlo@pw.edu.pl.*

REFERENCES

[1] Taylor, W. L., "Cloze procedure", J. Mass Commun. Quart., vol. 30, no. 4, pp. 415-433, 1953.

[2] Felice M., Taslimipoor S., Buttery P.. 2022. ``Constructing Open Cloze Tests Using Generation and Discrimination Capabilities of Transformers.``, Findings of the Association for Computational Linguistics (ACL 2022). Association for Computational Linguistics.

[3] Devlin J., Ming-Wei Chang, Lee K., Toutanova K.. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[4] Clark ./, Minh-Thang Luong, Quoc V. Le, Manning C.D., 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.

[5] Shlens, J., Notes on kullback-leibler divergence and likelihood theory, Systems Neurobiology Laboratory 92037 (2007): 1-4.

[6] liu, haozheng Zhang, Yiming, Meng, Qingdian, Zhang, Xu, Ccfbqgd: Chinese Cross-Domain Fill-in-The-Blank Question Generation Dataset and its Benchmarking Methodology. Available at SSRN: https://ssrn.com/abstract=4809985 or http://dx.doi.org/10.2139/ssrn.4809985

[7] Hu, Z., Chanumolu, R., Lin, X., Ayaz, N., Chi, V. (2021). Evaluating nlp systems on a novel cloze task: Judging the plausibility of possible fillers in instructional texts. arXiv preprint arXiv:2112.01867.

[8] Coniam, D. (1997). A preliminary inquiry into using corpus word frequency data in the automatic generation of English language cloze tests. Calico Journal, 15-33.

[9] 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, 9835-9850.

[3] This metrics for the test set is equal to 12.54.