Computer-Aided Methods for Intelligent Distractor Generation for Japanese Cloze Tests

Daniele Amore[1][a] and Michael Striewe[2][b]
[1]*Frankfurt University of Applied Sciences, Frankfurt, Germany*
[2]*Trier University of Applied Sciences, Trier, Germany*
daniele.amore@stud.fra-uas.de, m.striewe@inf.hochschule-trier.de

Keywords: Distractor Generation, Multiple-Choice Questions, Cloze Tests, Natural Language Processing, Japanese Language, Word Embeddings, BERT.

Abstract: This paper presents the development and evaluation of two systems for the automatic generation of distractors for Japanese cloze tests. The first system is a re-implementation of an existing architecture originally developed for Chinese, adapted to address the specific linguistic challenges of Japanese such as its complex writing system and agglutinative grammar. The second system extends this approach with a novel context-aware mechanism that leverages BERT to dynamically classify sentence contexts as open or closed and adjusts generation and filtering strategies accordingly. Both systems employ a two-stage pipeline of candidate generation and candidate filtering, utilizing criteria based on frequency similarity, orthographic similarity, word co-occurrence, and semantic similarity. Several inventories were constructed from a 25.4 million sentence Japanese Wikipedia corpus. The context-aware system further introduces a contextual similarity criterion and a BERT-based plausibility filter using Pseudo-Log-Likelihood scoring. A human evaluation study shows that both systems produce predominantly good or moderate distractors, with the word similarity and semantic similarity criteria yielding particularly strong results.

1 INTRODUCTION

Learning foreign languages is a cognitively demanding task whose success depends critically on the quality of teaching materials and the effectiveness of knowledge assessment. In the context of language acquisition, Multiple-Choice Questions (MCQs) serve as essential tools for evaluating language proficiency Alderson (2000). However, the effectiveness of an MCQ is crucially determined by the quality of its distractors, which are the incorrect answer alternatives designed to divert the test-taker from the correct solution (Susanti et al., 2018).

The manual creation of high-quality distractors is a demanding and time-consuming process (Liang et al., 2018). Automating this process through computer-aided methods offers the potential to significantly relieve human experts, such as teachers, in test creation and student assessment (Susanti et al., 2018).

While research and systems for automatic distractor generation already exist for languages such as English or Chinese (see, for example, Susanti et al. (2018) and Jiang and Lee (2017)), the Japanese language poses particular challenges due to its specific linguistic properties: a complex writing system consisting of Kanji, Hiragana, and Katakana; agglutinative grammar; and high context-dependency. Therefore, existing approaches cannot be directly transferred. This paper presents the design, implementation, and evaluation of two systems for automatic distractor generation for Japanese cloze tests. Section 2 introduces the foundational concepts underlying both systems, including statistical association measures, word embedding models, and pseudo-log-likelihood scoring. The first system (Section 3) is a re-implementation of the architecture proposed by Jiang and Lee (2017) for Chinese, adapted for Japanese. The second system (Section 4) is a novel Context-Aware Distractor Generation System (CADGS) that explicitly incorporates sentence context to dynamically adjust generation and filtering processes. Section 5 presents the evaluation results, and Section 6 concludes the paper.

[a] https://orcid.org/0009-0004-8806-8567
[b] https://orcid.org/0000-0001-8866-6971

2 BACKGROUND

2.1 MCQ Structure

In this work, a MCQ consists of a carrier sentence (a sentence with a gap), a target word (the correct answer to fill the gap), and several distractors (plausible but incorrect alternatives). Good distractors must be carefully selected as they need to be plausible yet lead to an incorrect answer (Jiang and Lee, 2017). More specifically, a good distractor should be *close* to the target word along one or more linguistic dimensions, such as similar corpus frequency (indicating comparable difficulty), shared orthographic features (e.g., common Kanji characters), frequent co-occurrence within sentences, or high semantic similarity in a vector space. This closeness is what makes the distractor appear plausible to the test-taker. However, a good distractor must simultaneously be distinguishable from the target word within the given sentence context, ensuring that it does not constitute a valid alternative answer. Achieving this balance between plausibility and incorrectness is the central challenge of distractor generation, and the criteria presented in Sections 3 and 4 each target different aspects of this closeness. Figure 1 illustrates an example MCQ structure.

Carrier Sentence: 私の____は、ニャーと鳴くととてもかわいいです。
(My ___ is very cute when it meows.)
Options:
1. 猫 (cat) ← *Target Word*
2. 犬 (dog) ← *Distractor*
3. 鳥 (bird) ← *Distractor*
4. 車 (car) ← *Distractor*

Figure 1: Example Multiple-Choice Question structure.

2.2 Statistical Association Measures

Pointwise Mutual Information (PMI) measures how much more frequently two words co-occur than expected under statistical independence:

$$\mathrm{PMI}(a,b) = \log \frac{p(a,b)}{p(a) \cdot p(b)} \tag{1}$$

A known weakness of PMI is its tendency to assign disproportionately high association strength to rare events. (Role and Nadif, 2011)

PMI^k. This variant introduces additional factors of $p(a,b)$ into the logarithm to empirically correct the bias toward rare events:

$$\mathrm{PMI}^k(a,b) = \log \frac{p(a,b)^k}{p(a) \cdot p(b)} \tag{2}$$

We use k=2 to achieve a moderate correction without over-emphasizing high-frequency but less informative word pairs.

2.3 Word Embedding Models

Word embedding models transform words into dense, low-dimensional vector spaces where semantic and syntactic relationships between words are represented by spatial proximity. These models are based on the distributional hypothesis, according to which words with similar meanings occur in similar contexts. (Johnson et al., 2023)

Word2Vec (Mikolov et al., 2013) uses shallow neural networks to learn vector representations for words. It offers two training architectures: Continuous Bag-of-Words (CBOW), which predicts the target word based on context, and Skip-gram (SG), which uses the target word to predict context words. (Al-Saqqa and Awajan, 2019)

FastText (Bojanowski et al., 2017) addresses Word2Vec's limitation of treating words as atomic units by additionally utilizing character n-grams. Each word is represented as a set of overlapping character sequences, and the word's vector representation is computed as the sum of its n-gram vectors. This approach is particularly beneficial for morphologically rich languages like Japanese, where a single verb can have over 100 inflected forms on average (Matsuzaki et al., 2024). Since many of these forms occur only rarely or not at all in the training corpus, no or only insufficient word embeddings can be learned for them. However, since most of these word forms follow clear morphological rules, their word embeddings can be improved through the incorporation of character-level information.

BERT (Ravichandiran, 2021) is a context-based embedding model that, unlike Word2Vec, generates different embeddings for the same word depending on its context. BERT uses a Masked Language Model (MLM) during training, where randomly selected words are replaced with a special `[MASK]` token, and the model learns to predict these missing words using the entire bidirectional context.

2.4 Pseudo-Log-Likelihood

BERT's bidirectional architecture does not permit direct computation of sentence log-likelihoods via the chain rule. Salazar et al. (2020) introduced Pseudo-Log-Likelihood (PLL) as an approximation:

$$PLL(W) := \sum_{t=1}^{|W|} \log P_{\mathrm{MLM}}(w_t | W_{\setminus t}; \Theta) \tag{3}$$

Each token w_t is individually masked and its conditional probability is computed given the full remaining context $W_{\setminus t}$. We employ the PLL-word-l2r variant proposed by Kauf and Ivanova (2023), which addresses the tendency of standard PLL to produce inflated scores for out-of-vocabulary words by masking all subsequent subword tokens within the same word during scoring.

3 RE-IMPLEMENTATION OF THE J&L ARCHITECTURE FOR JAPANESE

The architecture proposed by Jiang and Lee (2017) follows a two-stage pipeline: *candidate generation* produces a weighted list of distractor candidates using linguistic criteria, and *candidate filtering* removes candidates that could be interpreted as correct answers in the given sentence context.

3.1 Corpus Creation

Following the approach of Jiang and Lee, who used a 14-million-sentence Chinese Wikipedia corpus, we constructed a comparable corpus for Japanese from a full dump of the Japanese Wikipedia[1] (June 2025). Using WikiExtractor[2] for markup removal, a custom `WikiCorpusReader` class performs sentence segmentation based on Japanese punctuation marks and applies quality filters: sentences must be between 8 and 300 characters in length and consist of at least 30% Japanese script characters (Kanji, Hiragana, Katakana) to exclude non-Japanese text segments while preserving sentences containing legitimate foreign words. The resulting corpus[3] comprises 25.4 million sentences, exceeding the original Chinese corpus.

3.2 Candidate Generation

The candidate generation phase employs four independent criteria, each targeting a different aspect of similarity between the target word and potential distractors: frequency-based baseline matching, orthographic similarity, word co-occurrence, and semantic similarity. Each criterion produces its own ranked candidate list. All criteria share a common preprocessing foundation. A `CorpusProcessor` class tokenizes sentences using MeCab[4] with the UniDic dictionary and implements a *look-ahead logic* to address a key challenge of Japanese morphological analysis: conjugated verb and adjective forms are often split into atomic components by the analyzer. For example, the verb 食べた (tabeta, "ate") is split into the stem 食べ (tabe) and the auxiliary た (ta). Our look-ahead logic recombines these into single tokens, preserving semantic coherence. The result is a comprehensive *vocabulary inventory* storing, for each unique lemma (the base or dictionary form of a word, e.g., 走る for the inflected form 走った)–POS (Part-of-Speech, e.g., noun, verb, adjective) pair, its corpus frequency, observed surface forms, and detailed POS sub-categories. This inventory serves as the central data source for all subsequent generation criteria, enabling efficient lookup of word properties without reprocessing the corpus.

3.2.1 Baseline Criterion

Based on Coniam (1997), this criterion selects candidates that share the same part-of-speech (POS) as the target word and exhibit similar corpus frequency, approximating comparable difficulty levels. As illustrated in Figure 2, the target word is first analyzed by the `CorpusProcessor` to determine its lemma and POS. The vocabulary inventory is then filtered by POS, and candidates are ranked by ascending absolute frequency difference from the target word. A final cleaning step addresses an artifact of the MeCab dictionary, which occasionally appends English translations as suffixes to Katakana lemmas (e.g., チャット becomes チャット-time). Such suffixes are automatically detected and removed. This cleaning step is equally applied in the Spelling Similarity and Word Co-occurrence criteria.

3.2.2 Spelling Similarity Criterion

This criterion is based on the assumption that words sharing script characters are easily confused by learners (Jiang and Lee, 2017). As shown in Figure 3, the target word is first analyzed to obtain its lemma, POS, and corpus frequency. The individual Japanese script characters (Kanji, Hiragana, Katakana) are then extracted from the lemma of the target word. Functional characters such as the Chōonpu, a prolonged sound mark that extends the preceding vowel without carrying semantic meaning, are excluded from matching, as their high frequency across unrelated words would introduce excessive noise. Candidates are identified through an inverted index that maps each remaining

[1] https://dumps.wikimedia.org/jawiki/latest/

[2] https://github.com/attardi/wikiextractor

[3] https://zenodo.org/records/17195657

[4] https://pypi.org/project/mecab-python3/

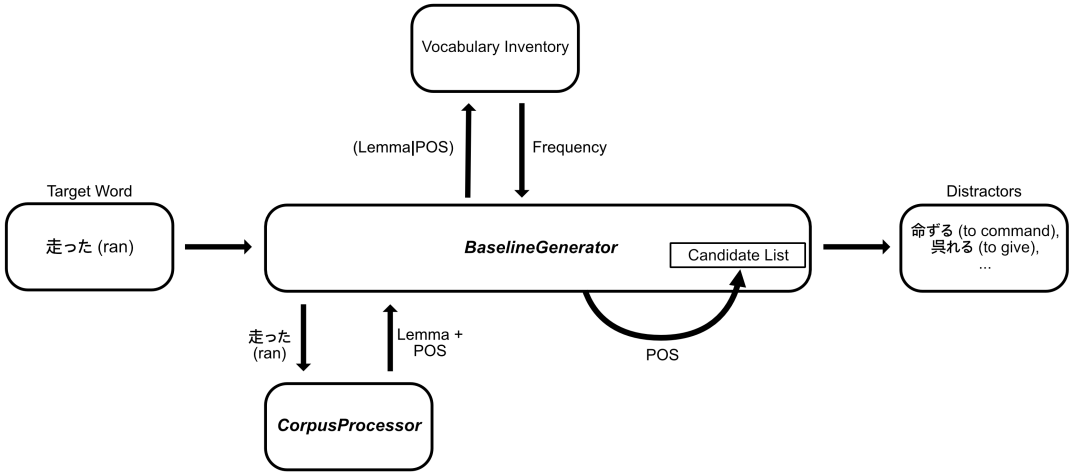

Figure 2: Generation process of the Baseline criterion.

character to all (lemma, POS) tuples in the vocabulary inventory containing that character. The resulting candidate set is filtered to retain only entries matching the target word's POS. The vocabulary inventory is queried twice during this process: first to retrieve the target word's corpus frequency, and then for each candidate's frequency. This allows candidates to be ranked by ascending absolute frequency difference, ensuring that distractors approximate the same difficulty level as the target word.

3.2.3 Word Co-Occurrence Criterion

This criterion is based on the hypothesis that a distractor frequently co-occurring with the target word may represent a plausible distraction for learners (Jiang and Lee, 2017). As illustrated in Figure 4, the target word is first analyzed by the `CorpusProcessor` to obtain its lemma, POS and corpus frequency. A co-occurrence inventory, constructed using a MapReduce pattern over the corpus, provides all words that co-occur with the target word within a sentence. The inventory is queried using the surface form of the target word first; if no co-occurring words are found, the lemma is used as a fallback. Co-occurrence is defined as the joint appearance of two unique tokenized words within a sentence; pairs occurring fewer than five times are removed, resulting in 58 million co-occurrence pairs. For each co-occurring word, the `CorpusProcessor` determines its lemma and POS, and candidates not matching the target word's POS are discarded. The vocabulary inventory is queried for each candidate's frequency to compute PMI (Equation 1). The strength of this association is used to rank the final candidate list.

3.2.4 Semantic Similarity Criterion

Words that are semantically similar to the target word tend to be plausible distractor candidates (Jiang and Lee, 2017). As shown in Figure 5, the target word's lemma is first determined by the `CorpusProcessor` as a fallback for out-of-vocabulary cases. The `SimilarityGenerator` then queries a Word2Vec model[5] to retrieve the most similar words. The model's `most_similar` function is first called with the surface form of the target word; if this fails, the lemma is used instead. Since word embedding models are based on the distributional hypothesis, according to which words occurring in similar contexts receive similar representations, and words of the same POS typically occur in similar syntactic contexts, it can be assumed that the generated candidates will largely preserve the target word's POS. Mikolov et al. (2013) demonstrated that their word embedding models capture both semantic and syntactic regularities of language, supporting this assumption. Therefore, no explicit POS filtering is applied. The returned candidate list is already ranked by descending cosine similarity, requiring no further sorting.

The underlying CBOW model (dimension 400, window size 5, minimum frequency 5, 5 training epochs) is trained on the Japanese Wikipedia corpus using Gensim[6]. Since Jiang and Lee (2017) did not specify the minimum frequency or epoch count, Gensim's default values were used. A crucial design decision is the removal of all particles from the training corpus. Japanese particles are high-frequency grammatical markers that appear in the context of countless semantically unrelated words, potentially caus-

[5]https://zenodo.org/records/17195922
[6]https://pypi.org/project/gensim/

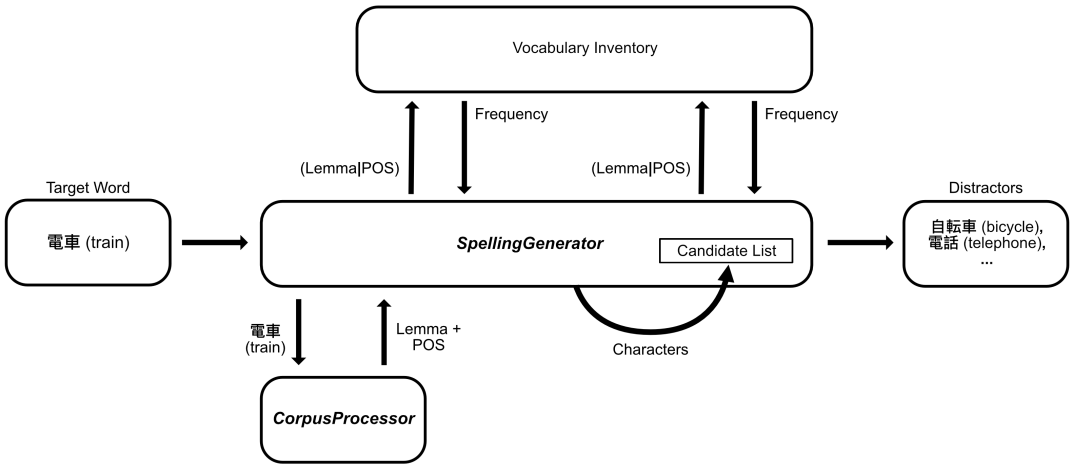

Figure 3: Generation process of the Spelling Similarity criterion.

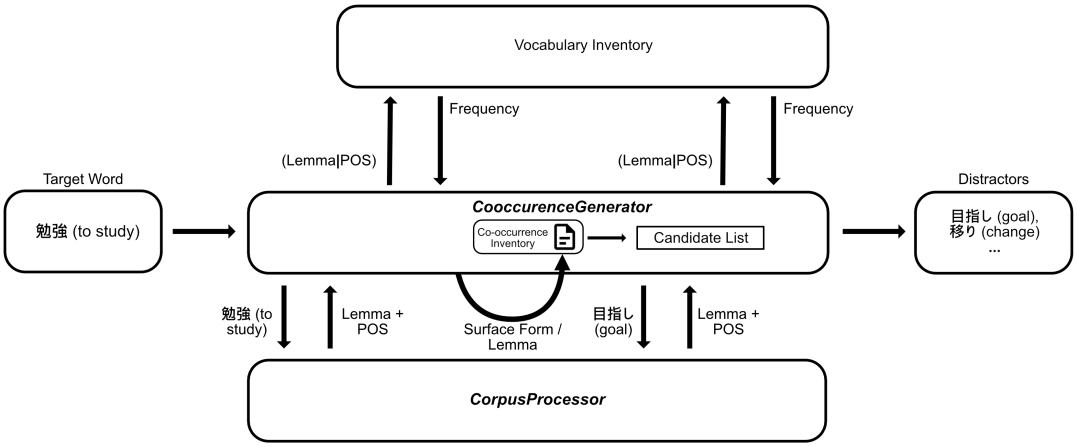

Figure 4: Generation process of the Word Co-occurrence criterion.

ing the model to learn syntactic rather than semantic patterns. By removing particles, the context window consists only of content words (nouns, verbs, adjectives), forcing the model to learn similarity based on shared thematic content rather than shared grammatical roles.

3.3 Candidate Filtering

3.3.1 Trigram Filter

This filter checks whether the word trigram formed by the distractor and its immediately preceding and following words in the carrier sentence occurs in the corpus. A trigram inventory is pre-constructed from the particle-free corpus to enable efficient lookup. If found, the candidate is considered too plausible and is removed. Operating on the particle-free corpus shifts the filter's function from syntactic to semantic plausibility checking.

3.3.2 Dependency Filter

To capture non-linear syntactic relationships beyond local trigrams, a dependency filter is implemented using the Ginza NLP framework[7]. An inventory of dependency triples (relation type, head lemma, child lemma) is extracted from the full corpus (including particles, as these are essential for correct syntactic analysis). Dependency relation types that primarily reflect grammatical function rather than semantic content, such as `case` and `aux`, are excluded from filtering. If a candidate forms a corpus-attested dependency relationship with another word in the carrier sentence, it is deemed too plausible and removed.

3.3.3 Adjusted Filter Logic

The original J&L architecture uses a logical AND, requiring both filters to flag a candidate for removal. Initial analysis revealed that the trigram filter exhibits

[7]https://github.com/megagonlabs/ginza

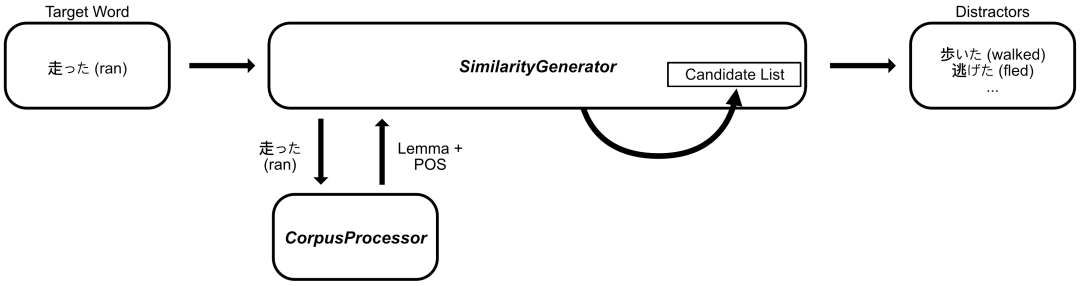

Figure 5: Generation process of the Semantic Similarity criterion.

very low filtering performance due to data sparsity on particle-free trigrams, while the dependency filter proves substantially more robust. To maximize distractor quality, we adopt a logical OR: a candidate is removed if *either* filter flags it.

4 CONTEXT-AWARE DISTRACTOR GENERATION SYSTEM

The re-implementation from Section 3 generates and filters candidates without considering the carrier sentence's context. This means the same distractors are produced regardless of whether the sentence strongly constrains the correct answer or leaves it largely open. As a consequence, distractors that are too close to the target word may become false positives in open contexts, while overly distant distractors may appear implausible in closed contexts. The Context-Aware Distractor Generation System (CADGS) addresses this limitation by explicitly incorporating the sentence context type to dynamically adjust thresholds and strategies.

4.1 Core Idea

To illustrate this, consider the MCQ from Figure 1 and a simplified variant of its carrier sentence: (A) "My ___ is very cute." and (B) "My ___ is very cute when it meows." With the target word 猫 ("cat"), a distractor like 犬 ("dog") would be a false positive in the *open* sentence (A), since "dog" is equally valid. In the *closed* sentence (B), however, "dog" becomes a good distractor because the context word "meow" restricts valid answers. This motivates context-dependent generation: in open contexts, distractors should be further from the target word; in closed contexts, they can be closer.

4.2 Context Type Classification

A `ContextAnalyzer` component determines the sentence context type. The carrier sentence with a `[MASK]` token is passed to BERT, which produces a probability distribution over its vocabulary for the masked position. The *entropy* of this distribution serves as the classification signal: high entropy (above a threshold of 4.5, determined through initial testing) indicates an open context where many words are plausible, while low entropy indicates a closed context where BERT is confident that only few words fit.

4.3 Adapted Candidate Generation

The Baseline and Spelling Similarity criteria remain identical to the re-implementation. The following criteria are adapted:

Word Co-occurrence. In open contexts, PMI^k (k=2, Equation 2) is used instead of standard PMI, shifting the ranking toward more general, less specific word pairs to reduce the risk of false-positive distractors. In closed contexts, standard PMI is used, as the restricted context already mitigates false positives.

Semantic Similarity. A pre-trained FastText model[8] replaces the custom-trained Word2Vec model, motivated by FastText's superior handling of out-of-vocabulary words through character n-gram representations. The similarity thresholds are dynamically adjusted based on context type: in open contexts, candidates with lower similarity scores are preferred; in closed contexts, higher similarity is permitted.

Contextual Similarity (New Criterion). This novel criterion leverages BERT's MLM capability to generate context-dependent distractors. As illustrated in Figure 6, unlike the previous criteria that find words similar to the target word in isolation, the `ContextualGenerator` receives both the carrier sentence with a `[MASK]` token and the target word as input. BERT predicts plausible words for the masked

[8]https://fasttext.cc/docs/en/crawl-vectors.html

position based on the full bidirectional context, producing a probability distribution over its vocabulary. Candidates are selected from a dynamically adjusted probability range: 0.5%–10% for open contexts and 0.5%–90% for closed contexts. The target word itself and incomplete subword tokens (prefixed with "##") are excluded from the candidate list.

4.4 BERT-Based Plausibility Filter

In addition to the trigram and dependency filters, the CADGS introduces a BERT-based filter using PLL-word-l2r (Kauf and Ivanova, 2023). For each candidate, the PLL score is computed for the carrier sentence with the candidate inserted. The PLL score of the sentence containing the target word serves as the reference, under the assumption that this sentence is definitively correct. A dynamic threshold, based on the reference PLL plus a context-dependent offset, determines whether a candidate produces a sentence that is too plausible (and should therefore be removed). For example, given the carrier sentence "In the café, I drink ___" with the target word コーヒー ("coffee", PLL $= -20.89$), the threshold computes to -22.14. A candidate like 緑茶 ("green tea", PLL $= -21.51$) exceeds this threshold and is removed as too plausible, while ケーキ ("cake", PLL $= -35.37$) falls well below and is retained as a valid distractor.

5 EVALUATION

5.1 Study Design

Two questionnaires were developed, one for each system, each containing 24 MCQs with 5–6 answer options. Each answer option was to be categorized by evaluators as: "Correct Answer" (target word), "Good Distractor," "Moderate Distractor," or "Poor Distractor." Both questionnaires use the same carrier sentences and target words; only the generated distractors differ.

The re-implementation questionnaire was evaluated by a native Japanese speaker and a student learning Japanese independently (≈N3 level). The CADGS questionnaire was evaluated by a different native Japanese speaker, a student of Japanese studies, and the same independent student.

5.2 Inter-Rater Agreement

Following Cohen's interpretation scale, both inter-rater agreement values indicate moderate agreement: Cohen's Kappa for the re-implementation questionnaire (2 raters) was $\kappa = 0.472$, and Fleiss' Kappa for the CADGS questionnaire (3 raters) was $\kappa = 0.41$. McHugh (2012) notes that values below 0.60 should be interpreted with caution. These results are comparable to the $\kappa = 0.529$ reported by Jiang and Lee (2017) for the same task.

5.3 Results

Table 1: Re-implementation evaluation results (Mod. = Moderate, Rel. = Reliability).

Criterion	Good	Mod.	Poor	Rel.
Baseline	12.5%	33.3%	54.2%	100%
Spelling	31.3%	37.5%	31.3%	100%
Similarity	56.3%	14.6%	18.8%	89.6%
Co-occurrence	4.2%	14.6%	81.3%	100%

Table 2: CADGS evaluation results (Mod. = Moderate, Rel. = Reliability).

Criterion	Good	Mod.	Poor	Rel.
Baseline	16.7%	29.2%	54.2%	100%
Spelling	38.9%	31.9%	29.2%	100%
Similarity	50.7%	6.8%	23.3%	80.8%
Co-occurrence	5.9%	26.5%	64.7%	97.1%
Contextual	28.1%	28.1%	35.9%	92.2%

Tables 1 and 2 present the results for both systems. "Reliability" indicates the percentage of cases where a distractor was *not* mistakenly identified as the correct answer; 100% means no distractor was ever chosen as the target word.

5.4 Discussion

It should be noted that the evaluation is based on a small number of raters (2–3 per system) and that both inter-rater agreement values are only moderate ($\kappa = 0.472$ and $\kappa = 0.41$). The results should therefore be interpreted as a preliminary, technically oriented proof of concept rather than a large-scale empirical study. Due to the anonymous design of the evaluation, a detailed analysis of how judgments differed between native speakers and language learners was not possible. Future work should consider non-anonymous evaluation designs to enable such comparisons and provide deeper insights into how proficiency level influences distractor perception. With this caveat in mind, the following patterns emerge from the data.

The **Semantic Similarity Criterion** consistently achieves the highest proportion of good distractors

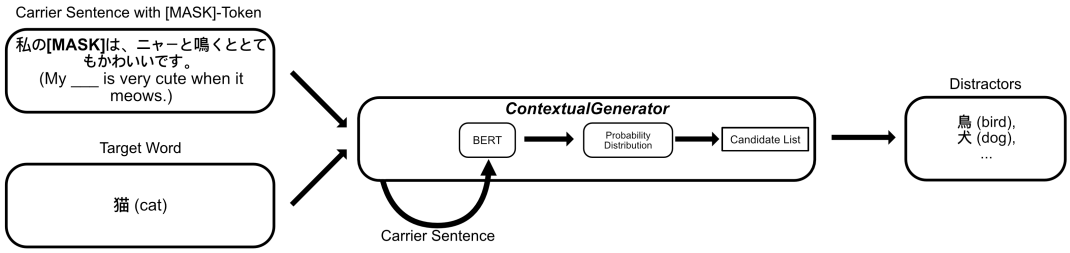

Figure 6: Generation process of the Contextual Similarity criterion.

in both systems (56.3% and 50.7%), demonstrating the effectiveness of the particle-free Word2Vec model and the FastText model for this task. However, this strong performance comes at the cost of reduced reliability (89.6% and 80.8%), as semantically close distractors are more likely to be valid answers.

The **Spelling Similarity Criterion** delivers solid and balanced performance, achieving 100% reliability while producing 31.3% (re-implementation) and 38.9% (CADGS) good distractors. This slight difference likely reflects evaluator variation, given that the Spelling criterion was not modified in CADGS, or highlights a positive effect of the new filter on distractor quality.

The **Co-occurrence criterion** shows the weakest performance overall but improves notably in the CADGS (poor distractors decrease from 81.3% to 64.7%), suggesting that the introduction of PMI^k for open contexts has a positive effect. One observed issue is that certain high-frequency, context-general words are disproportionately selected due to their broad co-occurrence patterns in Wikipedia.

The novel **Contextual Similarity criterion** of the CADGS shows balanced results (28.1% good, 28.1% moderate), providing a solid foundation for future refinement.

The switch from Word2Vec to FastText in the CADGS led to decreased reliability for the Similarity criterion (89.6% to 80.8%), suggesting that FastText's character n-gram representations produce candidates that are *closer* to the target word. While this confirms stronger semantic modeling, it indicates that the similarity thresholds require further optimization to maintain reliability.

The re-implementation proves overall more reliable. However, the CADGS outperforms the re-implementation in almost every criteria, additionally it has the potential to generate even better and more reliable distractors if the thresholds are optimized. Notably, the re-implementation's Similarity criterion achieves slightly lower reliability than Jiang and Lee's original system for Chinese, but substantially higher distractor quality (Jiang and Lee, 2017).

6 CONCLUSIONS

This paper presented two systems for automatic distractor generation for Japanese cloze tests. The re-implementation of the Jiang and Lee architecture for Japanese demonstrated that the two-stage pipeline of candidate generation and filtering can be successfully adapted to a morphologically complex language, with key innovations including a look-ahead logic for token recombination, particle removal for semantic model training, and an adjusted OR-based filter logic.

The Context-Aware Distractor Generation System extends this foundation with BERT-based context classification, dynamic threshold adjustment, a novel contextual similarity criterion, and PLL-based plausibility filtering. While the context-aware approach shows promising results, particularly for the co-occurrence and spelling criteria, further threshold optimization is needed to fully realize its potential.

Future work should address several areas: systematic optimization of CADGS thresholds, investigation of corpus effects (size, domain specificity), implementation of conjugation matching between target words and distractors, targeted Kanji prioritization in the spelling criterion, and evaluation across different proficiency levels (e.g., JLPT N5 vs. N3). Additionally, increasing the number and expertise of evaluators and conducting post-evaluation interviews would strengthen the empirical foundation.

ACKNOWLEDGEMENTS

The authors would like to thank all evaluation participants for their time and effort in rating the generated distractors. During the preparation of this manuscript, the authors used Claude[9] for language refinement and articulation throughout all sections of the paper. No content was generated by the AI system; it was used solely to improve the phrasing of the authors' original text. The authors reviewed and edited all output and

[9]Opus 4.5 and Opus 4.6 from Anthropic

take full responsibility for the content of this publication.

REFERENCES

Al-Saqqa, S. and Awajan, A. (2019). The use of word2vec model in sentiment analysis: A survey.

Alderson, J. C. (2000). *Assessing Reading*. Cambridge University Press, 1 edition.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information.

Coniam, D. (1997). A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests. *CALICO Journal*, 14.

Jiang, S. and Lee, J. (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. Association for Computational Linguistics.

Johnson, S. J., Murty, M. R., and Navakanth, I. (2023). A detailed review on word embedding techniques with emphasis on word2vec. *Multimedia Tools and Applications*, 83(13):37979–38007.

Kauf, C. and Ivanova, A. (2023). A Better Way to Do Masked Language Model Scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 925–935, Toronto, Canada. Association for Computational Linguistics.

Liang, C., Yang, X., Dave, N., Wham, D., Pursel, B., and Giles, C. L. (2018). Distractor generation for multiple choice questions using learning to rank. In Tetreault, J., Burstein, J., Kochmar, E., Leacock, C., and Yannakoudakis, H., editors, *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

Matsuzaki, K., Taniguchi, M., Inui, K., and Sakaguchi, K. (2024). J-UniMorph: Japanese Morphological Annotation through the Universal Feature Schema.

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, pages 276–282.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space.

Ravichandiran, S. (2021). *Getting Started with Google BERT: Build and Train State-of-the-Art Natural Language Processing Models Using BERT*. Packt, Birmingham Mumbai.

Role, F. and Nadif, M. (2011). Handling the impact of low frequency events on co-occurrence based measures of word similarity - a case study of pointwise mutual information.

Salazar, J., Liang, D., Nguyen, T. Q., and Kirchhoff, K. (2020). Masked language model scoring. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, H. (2018). Automatic distractor generation for multiple-choice English vocabulary questions. *Research and Practice in Technology Enhanced Learning*, 13(1):15.