Rupkatha Journal on Interdisciplinary Studies in Humanities (ISSN 0975-2935)
Indexed by Web of Science, Scopus, DOAJ, ERIHPLUS
Vol. 13, No. 2, April-June, 2021. 1-17
Full Text: http://rupkatha.com/V13/n2/v13n216.pdf
DOI: https://dx.doi.org/10.21659/rupkatha.v13n2.16
Published on May 12, 2021

Affix-based Distractor Generation for Tamil Multiple Choice Questions using Neural Word Embedding

Shanthi Murugan & Balasundaram S R

Department of Computer Applications, National Institute of Technology, Tiruchirappalli.

Email: shanthicse9@gmail.com, blsundar@nitt.edu

Abstract

Assessment plays an important role in learning and Multiple Choice Questions (MCQs) are quite popular in large-scale evaluations. Technology enabled learning necessitates a smart assessment. Therefore, automatic MCQ generation became increasingly popular in the last two decades. Despite a large amount of research effort, system generated MCQs are not useful in real educational applications. This is because of the inability to produce the diverse and human alike distractors. Distractors are the wrong choices given along with the correct answer (key) to confuse the examinee. Especially, in educational domain (grammar learning) the MCQs deal with affix-based or morphologically transformed distractors. In this paper, we present a method for automatic generation of affix-based distractors for fill-in-the-blanks for learning **Tamil Vocabulary**. Affix-based distractor generation relies on certain regularities manifest in high dimensional spaces. We investigate the quality of distractors generated by a number of criteria, including Part-Of-Speech, difficulty level, spelling, word co-occurrence, semantic similarity and affixation. We evaluated our proposed method in grammar based Multiple Choice Questions (MCQs) dataset. The result shows that affix-based distractors, yield significantly more plausible outcomes in certain grammar based questions.

Keywords: Multiple Choice Questions (MCQs), Assessment, affix-based distractors, grammar, automatic generation.

Introduction

The fill-in-the-blank item is a common form of exercise in Computer-Assisted Language Learning (CALL) systems. A fill-in-the-blank item is constructed on the basis of a carrier sentence. One word in the sentence is target word, or key which is blanked out, learner attempts to fill it. To enable automatic feedback from the learner, a fill-in-the-blank item often specifies choices, including target word and its several distractors. The Tamil example carrier sentence[1] is taken from Tamil 8th grade school book which is specified in Figure 1.

Figure 1

Tamil Fill-in-the-blank item from 8th grade Book

Fill-in-the-blank item: பண்டையமன்னர்கள்தமிழ் ______________ பாடுபட்டனர்

Correct Answer: வளர்ச்சிக்கு

Distractors: 1) வளர்ச்சியை 2) வளர்ச்சியின் 3) வளர்ச்சிகள்

Distractors are the wrong choices given along with the correct answer to befuddle the examinee. The quality of a fill-in-the-blank item largely depends on the quality of the distractors. If the distractors are not able to sufficiently confuse the examinee, the correct answer can be chosen easily. As a result, the overall quality and usability of the fill-in-the-blank item degrades.Distractors need to be carefully chosen: they must be sufficiently plausible, but must not be acceptable answers (Goodrich, 1977).

Generally, distractor generation approach recommends eight different distractor types for Multiple Choice Questions (MCQs), fill-in-the-blank items. Appropriately, affix-based distractors are suitable for grammar based fill-in-the-blank items rather than the remaining distractor types (Goodrich, 1977).In literature, among different types of distractors only few attempts are dealt with affix-based distractors (Aldabe et al, 2006) (Pino et al, 2009).In these approaches, affix-based distractors are generated from morphological generator (i.e.) NLP tools. But our method relies on high dimensional vector spaces to generate the affix-based distractors which is similar to morphological transformations work (Soricut et al, 2015).

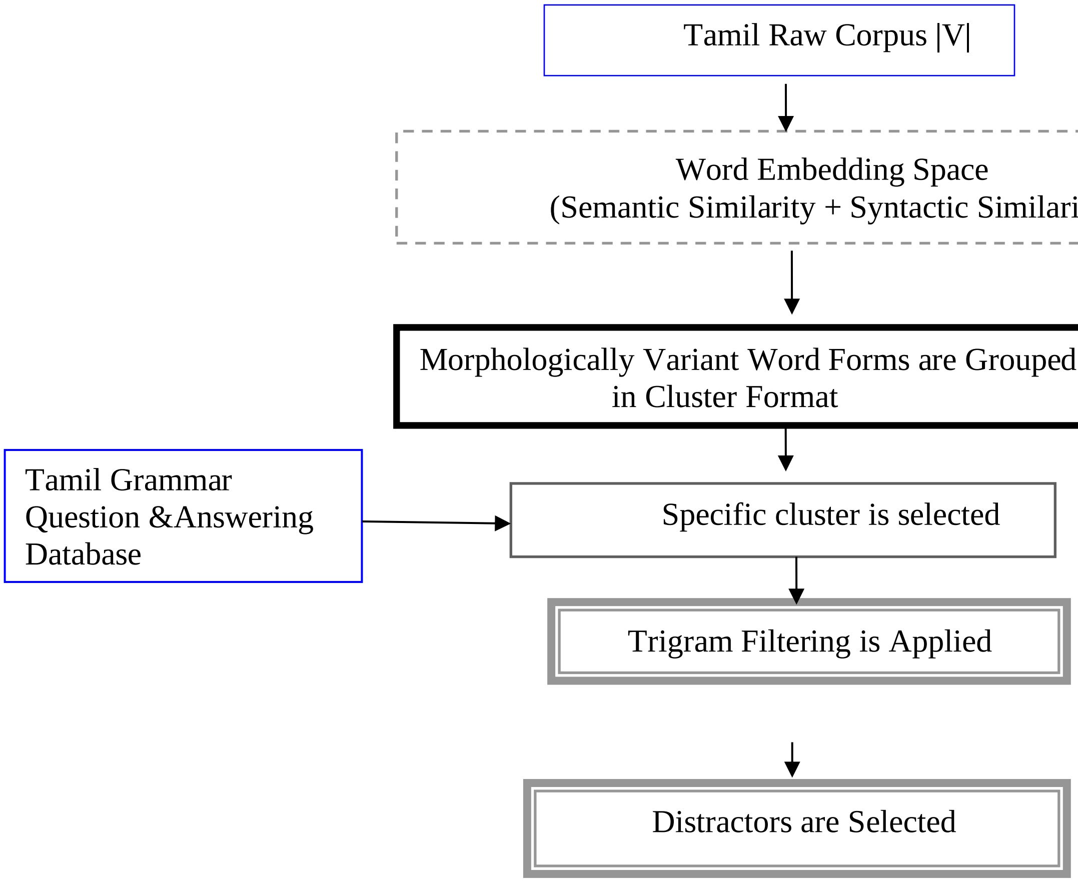
The present work primarily focuses on generating affix-based distractors for Tamil grammar questions which are practiced in 8th and 9th grade Tamil books as well as used in TNPSC group IV questions. Figure 1 illustrates the type of distractor which is discussed in this study. This type of Grammar questions are taken from school leveland TNPSC exams which are used to evaluate whether the student or candidate knows the Noun/Verb grammatical category details ((i.e.) Noun cases or verb Tenses). To-date, most research effort on distractor generation for language learning has focused on English. This paper presents the first attempt to **automatically** generate affix-based distractors for fill-in-the-blanks,MCQs of Tamil grammar questions in terms of Tamil language learning. Our proposed approach does not depend on annotated data like existing Tamil morphological generator.

Related Work

Goodrich (1977) uses two measures to evaluate eight type of distractors that were generated manually. Potency is the percentage of students who selects a specific choice (i.e) certain choices are not selected by examinee or frequently specific choices are selected by examinee. Next measurement is discrimination which is used differentiate the high proficiency and low proficiency students. We investigate affix-based distractorsamong eight type of distractorsin our study because they were suitable for grammar based fill-in-the-blanks. An Affix-based distractor is a word which is modified by morphemic addition or deletion.

Figure 2

Flow diagram of proposed work


Tamil Raw Corpus |V|
Word Embedding Space
(Semantic Similarity + Syntactic Similarity)
Morphologically Variant Word Forms are Grouped in Cluster Format
Tamil Grammar Question &Answering Database
Specific cluster is selected
Trigram Filtering is Applied
Distractors are Selected

Aldabe et al, (2006) designed a system to automatically generate questions and their distractors where distractors are obtained by morphological generator.They adapted the verb conjugation and noun declension tool for Basque language. They defined some of the parameters to generate the distractors such as changing the subject person, object person, verbal mode, tense, aspect, singular, plural, replacement and duplication of declension cases.Pino et al (2009) designed a system to semi-automatically generate incorrect choices, or distractors. They primarily focused on student’s misunderstanding type by generating the distractors. They used the XTAG system morphology database to generate morphological variant. Morph distractors are used to detect not only differences in morphological processing abilities which are useful to identify word integration skills.

Most existing methods for DG (Distractor Generation) are based on various similarity measures. These include Word Net-based metrics, Embedding-based similarities(Jiang and Lee, 2017), N-gram co-occurrence likelihood(Hill and Simha, 2016), phonetic and morphological similarities, Structural Similarities in an ontology, Context Similarity, Context Sensitive Inference, Syntactic Similarity(Chen et al., 2006), Word Difficulty Level(Frequency) and POS tags.

But our proposed work mainly focuses on affix-based distractor (i.e morphological variant with respect to correct answer or target word). Generally, affix-based distractors are generated from existing morphological generator which primarily rely on supervised learning mechanism and rule based systems. Supervised and rule based morphological generator mainly rely on linguistic rules and resources. Our method fully supports unsupervised learning for distractor generation. In word embedding space, certain regularities/similarities exist which are used to generate inflected form, singular, plural and its remaining categories with respect to target word of the fill-in-the-blank items. Word embedding regularities are used to generate possible number of word forms which are not specific with limited grammatical category. Certain regularities are: semantic regularities and syntactic regularities, example: king = men - woman +queen (semantic regularities). cars = vehicles – vehicle + car (syntactic regularities).

Our proposed work is similar to (Soricut et al., 2015) approach. In (Soricut et al., 2015) work, morphological variant forms are extracted based on word frequency. In our work, word length is considered for generating the affix-based distractors due to agglutinative nature of Tamil language. There is no free publicly available morphological generators for Tamil

Language.Existing morphological generator mainly depends on supervised machine learning(Rekha, R. U., et al,2010)as well as we need to specify the grammar category details (Morpho syntactic Information).There is no automated generation of morphological variant word for distractor generation. (Dhanalakshmi, Velliangiri, et al., 2010) specifies morphological generator system which is used as part of Tamil grammar teaching.

Affix-based Distractor Generation

We follow a three-step process where the **first is**, affix-based distractor candidate generation, that optimizes distractor plausibility; the **next one**, candidate filtering, that aims to filter out the distractor candidates and are acceptable answers to optimize distractor reliability.**last one** , candidate selection , that possibly selects the distractors from graph where the target word resides in graph. The flow of ourwork is depicted in figure2.

Affix-based Distractor Candidate Generation

Our approach hybrids the spell similarity and semantic similarity method for generating affix-based word generation. Our method describes following steps which are applied to Tamil monolingual training data over a finite vocabulary.

a. Train word embedding space on Tamil dataset for all words in vocabulary V.
b. Extract spelling similaritysuffix rules from vocabulary V
c. Evaluate quality of spelling similarity rules in word embedding space
d. Generate lexicalized morphological transformations

Train Word Embedding Space

Using large Tamil corpus, we train a word-embedding space E^n of dimensionality n for all words in |V| using the skip-gram model (Mikolov et al., 2013a) with negative sampling.

Extract Spelling Similarity Rules from Vocabulary V

The algorithm extracts all possible suffix rules (substitution) from word w_1 to word w_2 which has common stem. Morphological transformations rules are extracted from the entire vocabulary V words. Approximately, the rule size is (3 to 6).We denote such rules in the following form {from → to: { w_1:w_1’, w_2:w_2’, w_3:w_3’.., w_n: w_n’}}, where (from→ to) defines morphological transformations from word w_1 to word $w_{2,}$ w_1:w_1’ denotes word pair (word$_1$: word$_2$) ‘n’ number of word pairs { w_1:w_1’, w_2:w_2’, w_3:w_3’.., w_n:w_n’}in vocabulary V supports every extracted suffix rule or morphological transformation rule. To speed up the computation, we down sample the number of word pairs (50 word pairs have been used in the experiments)

For instance the rule {யை→யால்} is extracted from the set of word pairs where word pairs are belongs to vocabulary |V|.Morphological transformation rules (from→ to) are referred as candidate rules. The candidate rule set contains both rules which may **reflect true morphology phenomena** as well as such rules that simply **reflect surface-level coincidences.**

Evaluate Quality of Candidate Rules in Word Embedding Space

The extracted rule set reflects surface level coincidences as morphological transformations. These surface level coincidences do not preserve the meaningful morphological transformations. To identify as well as eliminate surface level coincidences from candidate rule set, we describe a generic evaluation function F. The extracted candidate rule setS_r = {from →to :{ w_1:w_1’, w_2:w_2’, w_3:w_3’.., w_n:w_n’}}.Evaluation function F is applied for every rule which is specified in equation 1. The direction vectors are calculated as specified in equation 2.

$$E_v^F ((w_1,w_2),(w,w')) = F_E(w_2,w_1 + \uparrow d_w) \quad (1)$$

$$(w_1,w_2),(w,w') \epsilon S_r \qquad \uparrow d_w = w'-w \quad (2)$$

Meaning preservation property of each rule in candidate set is evaluated by Equation 1 with the help of word pairs. We use a function F which defines cosine similarity rank function in embedding space. The cosine similarity rank function is calculated between vector value of *word*$_2$ and vector value of *word*$_1$ with addition of direction vector which is derived from remaining word pairs of the same rule. We can quantitatively measure the assertion “*காய்*”(vegetable**)** is to “*காய்கள்*”(vegetables) what “*மரம்*”(tree)is to “*மரங்கள்*”(trees), as rank function F (*காய்கள்***(**vegetables**)**, *காய்***(**vegetable)+*மரம்***(**trees – tree)).

We use a single threshold rank to capture meaning preservation (for the experiments in this paper use t’ rank =30): for each proposed rule r, we compute a hit rate based on the number of times Equation 1 scores above threshold rank, over the number of times it has been evaluated. We note that rules that are non-meaning-preserving receive low hit rates, while rules that are morphological in nature, which receives high hit rates. In Table 1 we present some of these candidate rules and their hit rate.

Table 1

Candidate rules are evaluated in terms of hit rate

Candidate Rule	Hit Rate
ஐ→ஆல்	80.09 %
க்கு→இன்	75.25 %
இல்→***இருந்து***	63.89 %
ஏன்→அது	53.62 %

Generate Lexicalized Morphological Transformations

Candidate rule does not taken into account for all words even though it satisfies meaning preservation constraint with high hit rate. Because, these candidate rules are applicable for specific set of lexicon as morphological transformations where the remaining lexicon candidate rule reflects surface level transformations. Morphological transformations need to be avoided when it become a meaningless transformations in surface level transformations.

Due to this condition, the rules are attached with specific group of lexicon along with direction vectors which preserve the meaningful transformations.From each rule best direction vectors are computed which explain the most pairs with the condition of Rank function(threshold value is 30) and Cosine Similarity function(0.50) in support set S_r. Iteratively pairs are extracted with direction vectors as subset from set S_r.

Graph Transformation: Morphological transformations are interpreted as graph based representation for the entire vocabulary |V| words. Words are represented as nodes, rules are act as edge between two nodes and rank, cosine values are noted as edge labels. We obtain N number of Multi-Directed graph G^{V}_{morph} for the entire vocabulary V which may contain cyclic nodes also.

Inducing 1-to-1 Morphological Mappings: We build a directed graph D^{V}_{morph} from G^{V}_{morph} as follows. Tamil language is agglutinative and morphologically rich language based on this property compared to prefix based transformations, suffix based morphemes are huge. Due to suffix consideration, normalizing the strongly connected multi directed graph G^V_{morph} to derive the directed graph D^V_{morph}based on **length property.**The sample graph is depicted in Figure 3.

We build a directed graph D^V_{morph} as follows:

a. edge$w_1 \rightarrow w_2$ in G^V_{morph} is considered only if length (w_1) >length (w_2).

b. if multiple such edges exist choose the one with minimal rank r

c. If multiple such edges still exist choose the one with the maximal cosine C.

Candidate Filtering

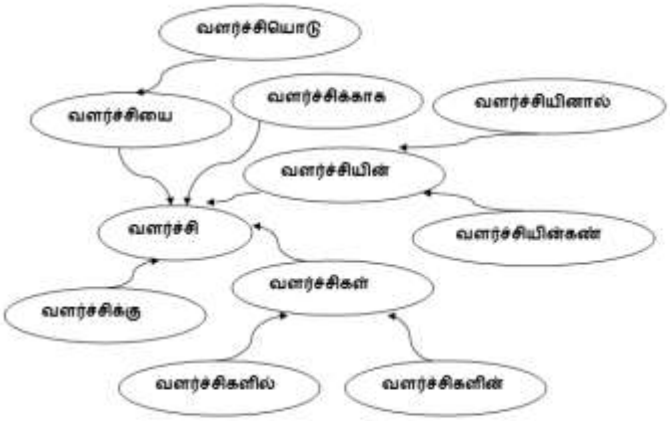
A distractor is called “reliable” if it yields an incorrect sentence. This step aims to remove those candidates that are also acceptable answers, leaving only the reliable distractors. We do so by examining whether the distractor can collocate with words in the rest of the carrier sentence. The system examines the distractor candidates generated from directed graph in candidate generation step, Trigram filter is used to examine the distractor candidate to maintain the reliability.

Trigram Filtering

The word trigram, formed by the distractor, the previous word and the following word in the carrier sentence, must not appear in the Wiki Corpus. For example the trigram “*தமிழ்வளர்ச்சிக்காகபாடுபட்டனர்*”found in the Wikipedia dump then the “*வளர்ச்சிக்காக*” is not considered as distractor. Similarly, words from the graph are matched in trigram property with wikipedia dump it is filtered out.

Figure 3

Morphological Family for the Normal- Form ‘*வளர்ச்சி* ’



Candidate Selection

The distractors are selected from graph where the correct answer or target word matches in the generated graph after candidate filtering. Randomly, distractors are selected in our experiments which are not specific with any approach. Due to random selection there is possibility to get more than one acceptable answer may be selected which reduces reliability of distractors.

In our work the question and its answer is given and its distractors need to be generated. The correct answer “*வளர்ச்சிக்கு*” matched with specific graph. The matched graph contains ‘n’ number of nodes which are considered as distractors. Randomly ‘N’ words are selected as distractors. Figure 4. shows possible correct answer and possible distractors. Possible correct answer are filtered by trigram filtering method when OOV(Out-of-Vocabulary) problem occurs then correct answer become a distractor because of random selection for distractors.

Fill-in-the-blank item: ***பண்டையமன்னர்கள்தமிழ்* ______________ *பாடுபட்டனர்***

Possible correct answer: 1) *வளர்ச்சிக்கு* 2) *வளர்ச்சிக்காக* 3) *வளர்ச்சியின்கண்*

Possible distractors: 1) *வளர்ச்சியை* 2)*வளர்ச்சியின்*3)*வளர்ச்சிகள்*

4)*வளர்ச்சியால்*5)*வளர்ச்சிகளில்* 6)*வளர்ச்சி*

Figure 4 : Fill-in- the-blanks with possible correct answer and its possible distractors

Data

To facilitate our experiments, we compiled following three datasets:

a. OPUS Corpus & MTIL Corpus: We extracted only Tamil monolingual data from parallel corpus to train the Tamil word Embedding[2].

b. Text Book Corpus: We manually extracted 200 grammar based Fill-in-the-blank questions from 8th & 9th grade Tamil text books. Extracted question distractor type is affix-based distractors.

c. Tamil Wikipedia Dump Corpus**:** We extracted sentences and word from Tamil Wikipedia dump[3] for calculating difficulty level (word frequency), spelling similarity, and word co-occurrence statistics. Trigram filtering also uses Wikipedia dump for candidate filtering.

Experiments

Baselines

The baseline system re-implements the criteria proposed by (Coniam, 1997): Generally distractor must have the same POS tag, similar difficulty level (frequency) with the target word. We also considered the spelling similarity method for generating distractors because they are partially similar to affix-based distractors. Co-occurrence method is used to generate the distractors but which are deviated from our affix-based distractor method. Semantic similarity from word$_2$vec method generates the distractors which is very similar to affix-based distractors if spelling is coincided with the target word otherwise semantic similarity method is not plausible.

Affix-based Distractor Generation

Initially, based on spell similarity and semantic similarity with the target word, distractors are generated as a forest (which is not specific to particular grammatical category). Spelling similarity methods are framed here as a candidate rule extraction. Semantic similarity method is adopted on the top of the spelling similarity method using evaluation function which is described in equation 1.Experimental details are represented in Table 2.In order to avoid more than one correct answer in distractor part, distractors are filtered using trigram filtering.

Table 2

Statistics Regarding the Size of the Training Data and the Induced Morphology Graphs

Language	Tamil
|Vocabulary Size|	5,20,000
|Number of Rules Extracted| (Spell Similarity)	58,000
|Number of Rules after Evaluation| (Semantic Similarity)	40,000
|Graph|	5000
|Directed Graph| (Affix-based Distractors)	3000

Evaluation and Results

Expert based Evaluation: In terms of Reliability and Plausibility the baseline system and our proposed system are evaluated.

We manually collected 200 grammar based questions with correct answer and their affix-based distractors in Tamil Language. For each of these 200 words, we generated distractors using five criteria from baseline model (POS tag, difficulty level (word frequency), spelling similarity, semantic similarity, word co-occurrence method).We asked two human judges; both are Tamil grammar teachers in school Level, to evaluate the choices which are generated by baseline and our proposed method.

Table 3

Reliability of the Various Distractor Generation Methods

Method	Reliability	Grammar based Questions
word co-occurrence	100 %	not suitable
POS tag	75%	partially suitable
spelling similarity	60%	partially suitable
semantic similarity	80%	partially suitable
proposed method (+spell & +semantic)	50%	Suitable

They further assessed the plausibility for distractors on a three point scale, "plausible" (3), "somewhat plausible"(2), or "obviously wrong"(1). Manually they assessed the reliability of a distractor which is generated by baseline and our proposed method. In Table 3 Reliability scores are shown and how well the baseline, proposed method distractors is suitable for grammar based questions.

Reliability (Distractor should Not be an Answer)

Word co-occurrence method is not suitable for generating distractors for grammar based questions. Semantic similarity, POS tag methods are suitable for generating distractors for grammar based questions when they implicitly coincide with spell similarity. Here, the generated distractors are possible to become a correct answer. Due to this, reliability score has been reduced. Candidate filtering eliminates more than one acceptable answer as a distractor with the reference of Wikipedia corpus for the given question. Spelling similarity method out performs in distractor generation for grammar based questions.

The grammar based question is used to evaluate the grammatical category for a particular stem or lemma (i.e) morphological variant or different inflected word form of the target word. Our proposed method outperforms in distractor generation compared to spelling similarity method.

Plausibility (similar to answer but not acceptable answer)

Our proposed method outperforms in distractor generation compared to semantic similarity, spelling similarity, POS tags. Semantic similarity method generates distractors which are semantically similar to correct choice but orthographically different. Spelling similarity generates the distractor which is orthographically similar but it might be semantically different. Our proposed approach integrates both spell and semantic wise similar distractors. In table 4 average score of plausibility is shown for distractors which are generated by baseline and our proposed method.

Table 4

Average Scores, Out of a 3-Point Scale Measures of Distractors Generated by the Various Methods in the Human Evaluation.

Method	Plausibility	Average Scores for the Plausibility
word co-occurrence	Not Plausible	1
POS tag	SomeWhat plausible	1.05
spelling similarity	SomeWhat plausible	2.25
semantic similarity	SomeWhat plausible	1.26
proposed method (+spell & + semantic)	Strongly plausible	2.58

Conclusion

We presented the first study on automatic generation of distractors for fill-in-the-blank items in Tamil grammar questions. Evaluations showed that a morphology based distractor which manifests in high dimensional vector space achieves competitive plausibility for Grammar based Fill-in-the-blank items in Tamil Language. As a future work, selection approach needs to be refined as a ranking method in candidate selection as well as candidate filtering methods needs to be refined over trigram filtering approach.

Notes

1.http://tnschools.gov.in/textbooks

2.http://opus.nlpl.eu/

3.https://dumps.wikimedia.org/tawiki/latest/

References

Aldabe, I., De Lacalle, M. L., Maritxalar, M., Martinez, E., & Uria, L. (2006, June). Arikiturri: an automatic question generator based on corpora and nlp techniques. In *International Conference on Intelligent Tutoring Systems* (pp. 584-594). Springer, Berlin, Heidelberg.

Chen, C. Y., Liou, H. C., & Chang, J. S. (2006, July). Fast–an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions* (pp. 1-4).

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.

Dhanalakshmi, V., Kumar, M. A., Rekha, R. U., Soman, K. P., & Rajendran, S. (2010, July). Grammar teaching tools for tamil language. In *2010 International Conference on Technology for Education* (pp. 85-88). IEEE.

Goodrich, H. C. (1977). Distractor efficiency in foreign language testing. *Tesol Quarterly*, 69-78.

Hill, J., & Simha, R. (2016, June). Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 23-30).

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Pino, J., & Eskenazi, M. (2009). Semi-automatic generation of cloze question distractors effect of students' l1. In *International Workshop on Speech and Language Technology in Education*.

Rekha, R. U., Kumar, M. A., Dhanalakshmi, V., Soman, K. P., & Rajendran, S. (2010, July). A novel approach to morphological generator for tamil. In *International Conference on Data Engineering and Management* (pp. 249-251). Springer, Berlin, Heidelberg.

Soricut, R., & Och, F. J. (2015). Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1627-1637).