Gap-fill Tests for Language Learners: Corpus-Driven Item Generation

Simon Smith
Xi'an Jiaotong Liverpool University, China
Simon.Smith@xjtlu.edu.cn

P.V.S Avinesh
IIIT Hyderabad, India
avinesh@research.iiit.ac.in

Adam Kilgarriff
Lexical Computing Ltd, UK
adam@lexmasterclass.com

Abstract

Gap-fill exercises have an important role in language teaching. They allow students to demonstrate that they understand vocabulary in context, discouraging memorization of translations. It is time-consuming and difficult for item writers to create good test items, and even then test items are open to Sinclair's critique of invented examples. We present a system, TEDDCLOG, which automatically generates draft test items from a corpus. TEDDCLOG takes the key (the word which will form the correct answer to the exercise) as input. It finds distractors (the alternative, wrong answers for the multiple-choice question) from a distributional thesaurus, and identifies a collocate of the key that does not occur with the distractors. Next it finds a simple corpus sentence containing the key and collocate. The system then presents the sentences and distractors to the user for approval, modification or rejection. The system is implemented using the API to the Sketch Engine, a leading corpus query system. We compare TEDDCLOG with other gap-fill-generation systems, and offer a partial evaluation of the results.

Key Words: gap-fill, Sketch Engine, corpus linguistics, ELT, GDEX, proficiency testing

1 Introduction

Gap-fill exercises are widely used throughout the language-teaching world. In a gap-fill (or cloze) test item, the student is presented with a text with one or more gaps in, and, for each gap, is asked to select the term that goes into it from a small number of candidates.[1,2] The tests allow targeted testing of particular competences in a controlled manner. Being multiple-choice, they are well-suited for automatic marking and are particularly useful in proficiency testing.

The standard method for producing test items is for an item writer to compose or locate a convincing carrier sentence, which incorporates the desired KEY (the correct answer, which has been deleted to make the gap). They then have to generate DISTRACTORS (wrong answers intended to 'distract' the student from selecting the correct answer). This is non-trivial as the distractor must be incorrect, in that inserting it into the blank generates a 'bad' sentence, yet the distractors must in some way be viable alternatives, or else the test item will be too easy.

The simple fact that carrier sentences are usually invented is also problematic. Since Sinclair (1986) the objections, in language teaching, to invented examples are well-established: it all too often occurs that invented examples do not replicate the phraseology and collocational preferences of naturally-occurring text.

TEDDCLOG (Testing English with Data-Driven CLOze Generation) is a system that generates draft test items using a very large corpus of English, using functions for finding collocates, distractors and carrier sentences in the Sketch Engine, a leading corpus query tool.[3] We use the

[1] The more widely-used name is cloze. However in some language-teaching literature, cloze is reserved for multi-sentence texts with several gaps to fill. Gap-fill is a more generic name.

[2] We focus here on multiple-choice exercises, though open gap fills have also been used (for example Pino et al, 2008).

[3] http://www.sketchengine.co.uk

Proceedings of ICON-2010: 8th International Conference on Natural Language Processing, Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2010

spectacular ukWaC freq = 34148

and/or	5623	1.7
panoramic	116	8.19
coastal	169	7.16
coral	30	6.46
unspoilt	28	6.39
rugged	34	6.38
scenic	31	6.15
mountainous	20	6.09
aerial	37	5.87
colourful	46	5.8
volcanic	23	5.78

adj_subject	2467	8.2
scenery	115	6.68
firework	41	6.18
sunset	13	4.92
nothing	149	4.17
reef	8	3.63
view	158	3.07
something	122	3.0
fall	12	2.64
anything	45	2.61
graphics	9	2.25

adj_comp_of	2892	5.3
look	181	3.46
sound	11	2.47
be	2473	1.2

modifies	23783	4.0
scenery	1295	9.52
sunset	183	7.41
view	3210	7.37
coastline	202	7.34
waterfall	158	7.24
backdrop	131	6.81
gorge	102	6.81
cliff	172	6.68
sight	299	6.48
display	527	6.48

modifier	2399	1.2
scenically	23	8.18
truly	327	7.25
visually	54	6.58
equally	73	5.53
pretty	100	5.51
quite	300	5.47
absolutely	43	4.63
similarly	12	4.38
suitably	9	4.36
utterly	11	4.36

Figure 1: Word sketch for showing collocates, grammatical relation, frequency and salience.

UKWaC corpus (Ferraresi et al 2008), a 1.5-billion-word web corpus. Ferraresi *et al* show that UKWaC is a good, broad sample of English. Size is important as then there are plenty of examples for most key-collocate pairings so the chances of finding a short, simple one suitable for language testing are high.

2 The System

The user inputs the key and its word class. Two Sketch Engine calls retrieve collocates and thesaurus items. We work through the two lists, checking each <collocate, thesaurus-entry> pair in turn to see if they co-occur in the corpus. We continue until we find a collocate and three thesaurus entries that do not occur with it.

An example: with the key *spectacular* the top collocates are *scenery, panoramic* (AND/OR relation) and *scenically*, as can be seen in the word sketch for *spectacular* (see Fig. 1).[4] The top thesaurus items are *stunning, magnificent, impressive* as shown in the thesaurus screenshot (Fig. 2). We take the first collocate and see if there are three thesaurus items that do not co-occur with it. (The corpus has been parsed at compile-time so this can be done quickly.) If we find three thesaurus items not occurring with the collocate, we are done. If not we move on to the next collocate and iterate. In this case we found three thesaurus items which were not found with the highest-scoring collocate, *scenery*.

[4]In the Sketch Engine and this work, a collocation is a triple involving two lemmas (of specific word class) and the grammatical relation holding between them; the grammatical relations applying here can be seen in Fig. 1.

spectacular ukWaC freq = 34148

Lemma	Score	Freq
stunning	0.545	35611
magnificent	0.508	28597
impressive	0.485	56146
dramatic	0.479	39157
amazing	0.461	61259
wonderful	0.448	91103
fantastic	0.444	74965
superb	0.435	51135
beautiful	0.431	123687
exciting	0.41	83964
splendid	0.384	15311

Figure 2: Distributional thesaurus entry.

The carrier sentence needs to contain *spectacular scenery*. There are 1295 such sentences in UKWaC. The next task is to choose the most suitable for a language-teaching, gap-fill exercise context.

TEDDCLOG uses the Sketch Engine's GDEX function (Good Dictionary Example Extractor (Kilgarriff et al 2008) to find the best sentence containing the collocation, choosing a sentence which is short (but not too short, or there is not enough useful contexts); begins with a capital let-

ter and ends with a full stop; has a maximum of two commas; and otherwise contains only the 26 lowercase letters. All others are rejected. These constraints may seem rigid, but in earlier tests we encountered many examples of sentences which were too technical or too informal to be comprehensible to learners. We found that excluding sentences that included symbols, numbers, quotation marks and proper names eliminated many of the problem items.

In our example, *spectacular* occurs in its base form. Sometimes verbs and adjectives occur in inflected forms in the carrier sentence, and in those cases we provide the key and distractors in the inflected form that the context requires.[5]

Next, we blank out the keyword, randomize the order of key and distractors to give a test item as here:

Some of the areas were high in the mountains where there was _____ scenery.

(a) historic

(b) spectacular

(c) huge

(d) exciting

---

3 System Evaluation

A random sample of 79 word-and-word-class pairs from the CEEC list[6] were entered into the system as the gap-fill item key. One carrier sentence and three distractors were generated for each key. The parameters used to identify candidate collocations were that the collocation's frequency needed to be greater than seven, and the salience[7] greater than zero. Test items were generated for 75 of the 79 input words.

The two authors of the paper who are native speakers of English (both linguists, and one an experienced language teacher) then assessed whether each item was acceptable or not, as a test item to be presented to intermediate English learners. They classified the cases where it was not.

For an item to be fully acceptable the carrier sentence must be:

- a well-formed sentence of English
- at a level that an intermediate learner of English can be expected to understand
- with sufficient context; not too short
- without superfluous material; not too long

Furthermore the distractors must be 'bad' but with some plausibility.

Results are given in Table 1. As we view TEDDCLOG as a drafting system, we have classified items according to whether they are acceptable as they are; acceptable after a minor edit (editing a maximum of one word); the carrier sentence is acceptable but one or more of the distractors are not; unacceptable.

Table 1: System Evaluation

Item action	**#**	**#**	**%**
Accept	40	53	
- As is	27		
- Edit carrier sentence	4		
- Change distractor(s)	9		
Reject		35	47
Total		75	100

3.1 Carrier sentence quality

Six items were not sentences. They were two noun phrases, two verb phrases, one adjective phrase, one containing the string *dh* (uninterpretable out of context) and one where the grammar was bungled.

Items where the language was grammatical but beyond the reach of most intermediate-level students included "Two muffins and a piece of rocky road and ____ latte?" (key=*skinny*, also a non-sentence), "Ursodeoxycholic acid normalises ____ and helps itching" (key=*biochemistry*) and "Consequentialism, in so far as it diverges from commonsense ____ on these points, strikes many as an unacceptable moral theory" (key=*morality*). The first case relates to informal vocabulary, the second, to specialist vocabulary, and the third, to formality of genre.

Items offering very little context included "These are followed by an alphabetical ____"

[5] The indefinite article can take two forms, a and an. If the carrier sentence contains an before the blank, it is obvious that the key must begin with a vowel. However, we do not wish to exclude consonant-initial distractors so if the indefinite article immediately precedes the key, we blank out both article and key, and offer article-noun pairs as fillers.

[6] A glossary of 6480 words used to help people studying for university entrance exams in Taiwan (see College Entrance Examination Center 2002).

[7] The salience statistic is based on the Dice coefficient; for details see 'Statistics used in the Sketch Engine' in the Sketch Engine help pages.

(key=*index*), "Of course, these are optical ____" (key=*illusions*) and "Then he noticed the ____ lever" (key=*gear*. There were no clear cases where sentences were too long (with the $morality$ example above being one of the longest).

One of the authors of the paper 'took the test' and was able to identify the correct answer in all but six of the 76 cases. Lack of context was not a problem for him because the compounds *alphabetical index, optical illusion* and *gear lever* are well-established. Here, the pedagogical issue becomes: what do we want to test for? For general knowledge of the meaning and use of the key, or specific knowledge of the compounds or other multi-word units it participates in? In cases like "Our family room was absolutely ____ ." (key=*superb*) the critical knowledge is of the collocation *absolutely superb*. In "Residents rushed to help and ____ the flames" (key=*smother*) the knowledge is of a non-core sense of the verb. In "Surely it is that injustice that will lead to ____ of discontent" (key=*a winter*) the knowledge is of Shakespeare.

3.2 Distractor quality

As we note above, for the test item to be satisfactory, the distractors must be bad, in the context of the carrier sentence. In each case, our algorithm guaranteed a collocation where the distractor did not occur in the corresponding construction in the corpus. A first concern is: can we expect the test-taker to know the collocation? A second is: does the absence of the distractor in the relevant construction indicate anything that we might expect the test-taker to know? If we look at "I had lain into the potato pie, mushy ____ and red cabbage" (key=*pea(s)*, distractors: *bean(s), spinach, carrot(s)*), do we wish to test knowledge of *mushy peas*? If not the test item does not work because *mushy beans/spinach/carrots* are all plausible. This was the norm in our dataset: getting the right answer depended on knowing a collocation including the key, and in the absence of that knowledge one or more distractors became a plausible answer.

There were several cases where particular distractors did not work for grammatical reasons. *speaks* has the wrong syntax to fill the gap in "The trouble is she always bloody ____ me" (key=*tells*). None of *access, download, manipulate* take a complement with *around* so only the key *navigate* fits in "Is it easy to ____ around the site?".

There was just one case where distractors were wrong enough to make the item notably easy: in "All rooms in the courts are fitted with a wash ____ basin" (key=*hand*), distractors *eye, body, head* are all very common core vocabulary and clearly do not fit.

4 Other gap-fill-generating systems

Several researchers have developed automatic gap-fill generators. Mostow et al (2004) generated gap-fill items of varying difficulty from children's stories. The items were presented to children via a voice interface, and the response data was used to assess comprehension. Hoshino & Nakagawa (2007) devised an NLP-based teacher's assistant, which first asks the user to supply a text. The system then suggests deletions that could be made, and helps the teacher to select appropriate distractors, chosen from among other words of the same class occurring in the same article, as well as their synonyms as recorded in WordNet. They also attempt to find distractors of approximately the same frequency as the key. In a teacher-user evaluation, 79% of the items generated were deemed appropriate.

Both of these systems use longer texts, while Sumita et al (2005) describe the automatic generation of single sentence gap-fill exercises from a large corpus. They use a published thesaurus to find potential distractors. To establish whether potential distractors are permissible in the carrier sentence, they submit queries to Google comprising the carrier sentence (or parts of it) with the key replaced by the potential distractor. They only retain distractors where Google does not find any hits.

To evaluate their system they gave tests items to a set of students for whom TOEIC English proficiency scores were known. They were able to show a high level of correlation between performance on the test items they had generated, and the students' proficiency scores. The correlation was similar to, but not as good as, the correlation between TOEIC scores and performance on expert-generated gap-fill items. A native speaker of English also did the test and scored 93.5%, higher than the highest-scoring non-native speaker who scored 90.6%.

For Liu et al (2005) the user input is word plus word sense. Much of their effort is spent on dis-

ambiguating the key in potential carrier sentences in order to find a carrier sentence in which the key is used in the intended sense. They succeed in doing this 65.5% of the time. Distractors are found in WordNet. The authors report 91.5% at generating sets of distractors which were not infelicitously correct.

Over several years the REAP project at Carnegie Mellon University has been developing web-based tools including gap-fill tests for learners of English. Recent work includes an investigation of distractors based on morphological (*boring* vs. *bored*), orthographic (*bread* vs. *beard*) and pronunciation-based (*file* vs. *fly*) confusability (Pino & Eskenazi, 2009). Their gap-fill generation system (Pino et al, 2008) explores using examples from WordNet, from a learners' dictionary, and from a large corpus of documents suitable for learners, as carrier sentences. They identify good sentences by assessing complexity, well-defined context, grammaticality and length. They look at expert-produced test items to find optimum structures and lengths. They use the Stanford parser to assess complexity, and whether a sentence is grammatical. They assess whether there is a well-defined context by computing the extent to which the words in the sentence associate with each other based on the pointwise mutual information of the word pairs. If the words cohere, in this sense, that implies a well-defined context. They then also use this framework for identifying potential distractors: they are words which fit quite well but not too well in the context. In their evaluation, 66.5% of questions were found to be acceptable, with the most common flaw being that some of the distractors were acceptable.

5 Critical comparison with other systems, and future work

The systems reviewed offer a number of insights that we plan to integrate into TEDDCLOG in the future.

A first point is sentence length. Liu et al analyse a batch of expert-generated test items and find the mean length to be 16 words. In our dataset the average was eleven. As our internal evidence also indicates, GDEX parameters need adjusting to favour longer sentences giving more context. Further structure is given to the theme by Pino et al's (2008) observation that high-quality carrier sentences often consist of two clauses, one containing the key and the other specifying the context (though this could be a feature of the fact that this is how these invented sentences are constructed, so could fall foul of the 'inauthenticity' critique).

Our method focuses on a single collocation of the key's. There is often nothing to favour key over distractors except knowledge of the compound or collocation. If the goal of the exercise is to test knowledge of the core meaning of the word (as opposed to knowledge of its collocation, a more advanced topic) then such test items fail. Pino et al and Liu et al look at the overall coherence of the candidate sentence by checking for relations between all the content words and each other. This is a technique we shall add, both for finding coherent sentences (where the key is in place) and for strengthening the evidence that a distractor is bad.

Sumita et al's approach to negative evidence is via Google. While this is appealing, it has a downside: Google places limits on the number of queries allowed per day, it is not clear what a suitable length of sentence fragment to submit to Google is, and replicability is lost (Kilgarriff, 2007). We prefer the model in which very large corpora (compiled from the web) are used; we shall soon start using a corpus of 5 billion words. Then, measuring coherence between all words and the distractor will also give us more evidence (including negative evidence) in relation to the question "is a distractor accidentally acceptable?"

As a source for carrier sentences, Pino et al use a database of pre-selected web texts. This is similar to our approach except that we first gather very large numbers of web texts, and include them all in the corpus. We then select suitable sentences at run time, using GDEX. We have also been exploring pre-classifying all documents for readability and including that information in the document header (which is accessible to the search tools at run time).

Mostow et al and Hoshino and Nakagawa's systems start from texts or sentences, whereas we, like Liu et al, start from the key. This is significant for two reasons: first, because item writers generally wish to use a specific word as a point of departure for producing a gap-fill item. Second, our architecture is capable of generating large numbers of gap-fill items on a given topic (*Business*, perhaps, or *Starting out at University*).

Currently we use the Sketch Engine's shallow

parser, which supplies grammatical relations between pairs of words but no bracketing. We intend to follow Sumita, Liu and Pino in using a fuller parser, and exploiting its output to find sentences of suitable grammatical structure.

We are planning to explore frequency factors further. Mostow et al are among the authors proposing distractors of similar frequency to the key. We do this indirectly, to some extent, as words tend to be classified as similar to other words of similar frequency in a distributional thesaurus such as the Sketch Engine's (Weeds and Weir, 2005). Currently we set the frequency threshold for the collocation involving the key at just seven. We get some obscure collocations like *umeboshi plums*. We plan to explore setting this threshold much higher.

We are impressed by Sumita et al's evaluation, and in particular the way it addresses the questions "what level of difficulty in the text is acceptable? How subtle are the contrasts between key and distractors allowed to be?" They evaluate by correlating with students' TOEIC scores. This allows that some test items are hard (and only the best students will get them right), others are intermediate, and others are easy (so the less good students will often get them right). Particularly for proficiency testing, it is often convenient to have questions at a range of levels.

Perhaps the most important way forward is to look more closely at the different competences that gap-fill tests are used to test, if possible in consort with professional testers, and to tailor our algorithms to their particular tasks. This is what we intend to do. In that context, we shall offer item-writers a number of carrier sentences and distractors, to choose from and edit for each key.

6 Summary

We have described a program which generates gap-fill exercises with distractors which will appear, to many students, to be plausible correct answers. Using a very large corpus, and methods from computational linguistics, TEDDCLOG offers the prospect of making the preparation of gap-fill items (currently labour-intensive and vulnerable to the 'invented example' objection) both faster and based on real language data.

References

College Entrance Examination Center, High School English Reference Wordlist, Retrieved December 29, 2008, from `http://www.ceec.edu.tw/Research/paper_doc/ce37/ce37.htm`

Ferraresi, A., Zanchetta, E., Bernardini, S. & Baroni, M. 2008, *Introducing and evaluating ukWaC, a very large web-derived corpus of English.* Proceedings, 4th WAC workshop, LREC, Marrakech, Morocco.

Hoshino, A. and Nakagawa, H. 2007, *Assisting cloze test making with a web application.* Proc. Society for Information Technology and Teacher Education International Conference 2007, (pp. 2807-2814). Chesapeake, VA: AACE.

Kilgarriff, A. 2007, *Googleology is bad science.* Computational Linguistics 33(1):147-151.

Kilgarriff, A., Husak, M., McAdam, K., Rundell, M. and Rychlý, P. 2008, *GDEX: Automatically finding good dictionary examples in a corpus.* Proc. EURALEX, Barcelona.

Liu, C-L., Wang C-H & Gao Z-M. 2005, *Using lexical constraints to enhance computer-generated multiple-choice cloze items.* International Journal of Computational Linguistics and Chinese Language Processing 10(3), 303-328.

Mostow, J., Beck, J. E., Bey, J., Cuneo, A., Sison, J., Tobin, B. and Valeri, J. 2004, *Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions.* Technology, Instruction, Cognition and Learning 2: 97-134.

Pino, J. and Eskenazi, M. 2009, *Semi-Automatic Generation of Cloze Question Distractors Effect of Students' L1.* SLaTE Workshop on Speech and Language Technology in Education.

Pino, J., Heilman, M. and Eskenazi, M. 2008, *A Selection Strategy to Improve Cloze Question Quality.* Wkshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th Int. Conf. on ITS.

Sinclair, J., ed. 1986, *Looking Up: An Account of the COBUILD Project in Lexical Computing.* Collins COBUILD, London and Glasgow.

Sumita, E., Sugaya, F. and Yamamoto, S. 2005, *Measuring Non-native Speakers Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions.* 2nd Wkshop on Building Educational Applications using NLP, Ann Arbor.

Taylor, W. L. 1953, *Cloze procedure: A new tool for measuring readability.* Journalism Quarterly 30: 415-433.

Weeds, J., and Weir, D. 2005, *Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity.* Computational Linguistics 31:4.