Predicting the Difficulty of Language Proficiency Tests

Lisa Beinborn[◇‡], **Torsten Zesch**[§] **Iryna Gurevych**[◇‡]

◇ UKP Lab, Technische Universität Darmstadt
‡ UKP Lab, German Institute for Educational Research
§ Language Technology Lab, University of Duisburg-Essen
http://www.ukp.tu-darmstadt.de

Abstract

Language proficiency tests are used to evaluate and compare the progress of language learners. We present an approach for automatic difficulty prediction of C-tests that performs on par with human experts. On the basis of detailed analysis of newly collected data, we develop a model for C-test difficulty introducing four dimensions: solution difficulty, candidate ambiguity, inter-gap dependency, and paragraph difficulty. We show that cues from all four dimensions contribute to C-test difficulty.

1 Introduction

In a labor market that is increasingly globalized, knowledge of at least one foreign language is more relevant than ever before. Due to increased mobility, multilingual skills are also required for private communication as friendships stretch across geographical and linguistic borders. In order to provide adequate language learning support, it is important to frequently evaluate learner progress on the basis of language proficiency tests that enable a fair comparison between learners.

The test difficulty needs to match the intended target group as the test should be challenging for the learner but not lead to frustration. According to Vygotsky's zone of proximal development (Vygotsky, 1978), the range of suitable material is very small. Thus, creating a test that fits this narrow target zone is a tedious and time-consuming task. Teachers predict the difficulty of a test based on their teaching experience. However, as they already know the solutions, they cannot always anticipate the confusion a test might cause for learners. This results in a subjective difficulty estimation that often lacks the consistency required for comparing learners over different tests.

The underlying principle of most language proficiency tests is the concept of reduced redundancy testing (Spolsky, 1969). It is based on the idea that "natural language is redundant" and that more advanced learners can be distinguished from beginners by their ability to deal with reduced redundancy. For language testing, redundancy can be reduced by eliminating words from a text and asking the learner to fill in the gap, also known as the cloze test. The C-test is a variant of the cloze test which contains more gaps but provides part of the solution as a hint and has been found to be a good estimate for language proficiency (Eckes and Grotjahn, 2006).

We present an approach for determining the difficulty of C-tests that overcomes the mentioned drawbacks of subjective evaluation by teachers. Our approach is based on objective measurable properties and thus produces consistent results. We show that our approach performs on par with human experts and analyze to which extent C-test difficulty is determined by individual gap properties (micro-level processing) and higher level dependencies (macro-level processing). On the theoretical level, our model provides new insights into the factors that affect difficulty in reduced redundancy testing. On the practical level, our results may help teachers to precisely evaluate the difficulty of a test and to foresee challenging parts.

2 The C-Test

The C-test is a form of reduced redundancy testing and has been established as a standard entrance exam for many language centers. It usually consists of five coherent paragraphs or short texts. The example below consists of a single paragraph.

> *The roots of humanity can be traced back to millions of years ago. T___ primary evid___ comes fr___ fossils - skulls, skel___ and bo___ fragments. Scien___ have ma___ tools th___ allow th___ to ext___ subtle infor___ from anc___ bones a___ their enviro___ settings. Mod___ forensic wo___ in t___ field a___ in labora___ can n___ provide a rich understanding of how our ancestors lived.*[1]

[1] Solutions: *The, evidence, from, skeletons, bone, Scientists, made, that, them, extract, information, ancient, and, environmental, Modern, work, the, and, laboratories, now*

Transactions of the Association for Computational Linguistics, vol. 2, pp. 517–529, 2014. Action Editor: Chris Callison-Burch.
Submission batch: 4/2014; Revision batch 8/2014; Published 11/2014.

After an unaltered introductory sentence, every second word is transformed into a gap. When the intended number of gaps is reached (usually 20), the rest of the text is left intact. For each gap, the smaller half of the word is provided and the missing part has to be completed by the learner. Since its introduction, the C-test has been researched from many angles and has been adapted for over 20 languages (see Grotjahn et al. (2002) for an overview).

2.1 C-Tests vs Cloze Tests

The *C* in C-test stands for its origin in the cloze test. In cloze tests, full words are transformed into gaps according to a fixed deletion pattern (e.g. every 7th word).

The main problem with cloze tests is the ambiguity of the solution. Unless function words are deleted, the gap allows many alternative solutions such as synonyms and hypernyms, but also entirely different words that change the meaning of the text but also fit the context. Language teachers have proposed two ways of dealing with this ambiguity: the application of relaxed scoring schemes and the use of distractors. In relaxed scoring, teachers accept all tolerable candidates for a gap and not only the intended solution as in exact scoring. Unfortunately, this scoring method turned out to be quite subjective and time-consuming as it is not possible to anticipate all tolerable solutions (Raatz and Klein-Braley, 2002). The use of distractors circumvents this open solution space by providing a closed set of candidates from which the solution needs to be picked. Several approaches have been proposed for automatic distractor selection (Sakaguchi et al., 2013; Zesch and Melamud, 2014) to make sure that the distractors are not too hard nor too easy and are not a valid solution themselves. However, the presence of the correct solution in the distractor set enables the option of random guessing leading to biased results.

In order to overcome this and other weaknesses of the cloze test, Klein-Braley and Raatz (1984) propose the C-test as a more stable alternative. Thorough analyses following the principles of test theory indicate advantages of the C-test over the cloze test regarding empirical validity, reliability, and correlation with other language tests (Babaii and Ansary, 2001; Klein-Braley, 1997; Jafarpur, 1995). For automatic approaches, the following properties of the C-tests are beneficial: The given prefix restricts the solution space to a single solution (in almost all cases) which enables automatic scoring without providing a guessing option. In addition, the prefix hint allows for a narrower deletion pattern (every second gap) providing more empirical evidence for the students' abilities on less text.

As the given prefixes reduce the extent to which productive skills are required, Cohen (1984) considers the C-test to be a test of reading ability examining only recognition. However, Jakschik et al. (2010) transform the C-test into a true recognition test by providing multiple choice options and find that this variant is significantly easier than open C-test gaps. This indicates that C-test solving requires both, receptive and productive skills, and we reflect this in our feature choice.

2.2 Test Difficulty

Previous works in the field of educational natural language processing approach language proficiency tests from a generation perspective. The focus is on generating closed formats such as multiple choice cloze tests (Mostow and Jang, 2012; Agarwal and Mannem, 2011; Mitkov et al., 2006), vocabulary exercises (Skory and Eskenazi, 2010; Heilman et al., 2007; Brown et al., 2005) and grammar exercises (Perez-Beltrachini et al., 2012). The difficulty of these exercises is usually determined by the choice of distractors as students have to discriminate the correct answer from a provided set of candidates.

C-tests follow a fixed construction pattern and are therefore easy to generate. As opposed to closed formats, the candidate space is only limited by the provided prefix and the length constraint. It is thus harder to determine the difficulty of a C-test because it is influenced by a combination of many text- and word-specific factors. The search for the factors that determine the difficulty of C-tests is tightly connected to the question of construct validity: "Which skills does the C-test measure?" While advocates of the C-test argue that it measures general language proficiency involving all levels of language (Eckes and Grotjahn, 2006; Sigott, 1995; Klein-Braley, 1985) others reduce it to a grammar test (Babaii and Ansary, 2001) or rather a vocabulary test (Chapelle, 1994; Singleton and Little, 1991).[2] In our model, we aim at combining features touching all levels of language. The earliest analyses of C-test difficulty focused on the paragraph instead of the gap level. Klein-Braley (1984) performs a linear regression analysis with only two difficulty indicators – average sentence length and type-token ratio – obtaining good results for her target group. Eckes (2011) intend to calibrate C-test difficulty using a Rasch model in order to compare different C-tests and build a test pool.[3]

Kamimoto (1993) was the first to perform classical item analysis on the gap level. He created a tailored C-test that only contains selected gaps in order to better discriminate between the students. However, the gap selection is based on previous test results instead of specific gap features and thus cannot be applied on new tests.

Previous work on gap difficulty is based on correlation analyses. Brown (1989) identifies the word class, the local word frequency, and readability measures as factors correlating with cloze gap difficulty. Sigott (1995) exam-

[2] It should be noted, that their definition of "vocabulary" is very wide.

[3] http://www.ondaf.de

	T1	T2	T3	T4
Participants	357	156	147	160
Mean error rate	.31	.46	.37	.36
Standard deviation	.21	.26	.24	.28

Table 1: Analysis of text-level test difficulty

ines word frequency, word class, and constituent type of the gap for the C-test and finds high correlation only for the word frequency. Klein-Braley (1996) identifies additional error patterns related to production problems (right word stem in wrong form) and early closure, i.e. the solution works locally but not in the larger context. The cited works focus on the correlation between gap features and C-test difficulty but did not attempt to actually predict difficulty. In the following section, we present the results of our data analysis targeted towards building up a model for C-test difficulty.

3 Data Analysis

For a better understanding of C-test difficulty, we need to perform data analysis. As suitable data was not available in digital form, we conducted a data collection study. In cooperation with the language center at Technische Universität Darmstadt, we gathered data from 3 test sessions. The C-tests are conducted in order to assign students to courses matching their language proficiency. One test consists of 5 paragraphs with 20 gaps each.

We created a web interface in which the test had to be filled. Most students finished before the time limit of 20 minutes was reached. Weaker students left some gaps unfilled but did not ask for more time. In the first session, 357 participants filled in the same C-test (T1). In the second session, three different test instances (T2, T3, T4) were assigned randomly to 463 new participants. In the third session, the tests were composed by randomly choosing paragraphs from 5 groups, each consisisting of 5 paragraphs. A random combination of 5 paragraphs (one from each group) was then assigned to 1050 new participants. All participants are students enrolled at the university. Our analysis is based on the first two sessions and we use the data from the third session as test data. As six paragraphs of the third session had already been administered before, we remove these from the test data.

3.1 Text-level Analysis

As C-tests are designed mainly for the goal of comparing students, the difficulty of different tests should be balanced. The difficulty of a C-test is usually measured by the mean error rate over all gaps. The error rate of a single gap is the ratio of false answers to all answers. A higher mean error rate thus indicates higher test difficulty.

As we see in Table 1, the mean error rate varies between the different tests, although they had been carefully

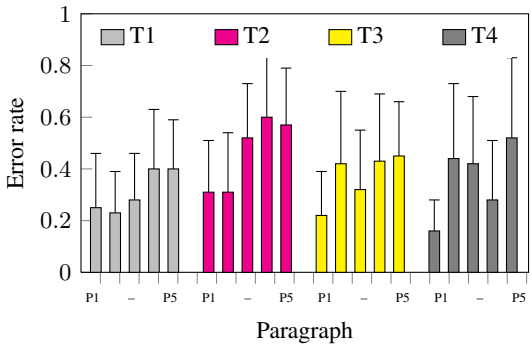

Figure 1: Mean error rate and standard deviation for the paragraphs 1–5 of the four tests

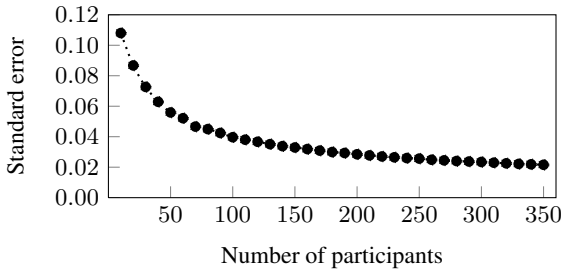

Figure 2: Standard error averaged over all gaps for increasing numbers of participants

(but manually) designed to be equally difficult. The first session was generally easier than the second session, and T2 stood out as particularly difficult. Within this test setting, it is thus not fair to compare students by their overall score, if they completed different tests. Automatic difficulty prediction prior to the test session could improve the comparability of test results.

Figure 1 additionally shows the results for each paragraph. The teachers arrange the five paragraphs of a test with assumed ascending difficulty. We see that this works as a general tendency (paragraph 5 is more difficult than paragraph 1), but a true ordering has not been achieved for any test. In general, the high standard deviations indicate that the mean error rate is not a very informative measure, because each test contains very easy and very difficult gaps. In the extreme case, half of the gaps can be solved by all learners and the other half by almost no one. The test is then assigned a medium difficulty, but the results are not useful for discrimination between learners. We therefore now analyze the difficulty on the gap-level.

3.2 Gap-level Analysis

Before we can further analyze single gaps, we need to examine whether the number of participants in our study was sufficient to obtain reliable error rates on the gap level. We calculate the standard error for each gap with

The roots of humanity can be traced back to millions of years ago. T primary evid comes fr fossils - skulls, skel and bo fragments. Scien have ma tools th allow th to ext subtle infor from anc bones a their enviro settings. Mod forensic wo in t field a in labora can n provide a rich understanding of how our ancestors lived.

Figure 3: Visualisation of error rates for each gap

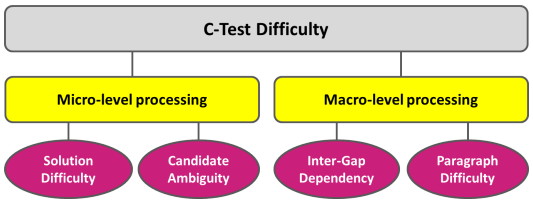

Figure 4: C-Test Difficulty Model

increasing sample sizes.[4] Figure 2 shows the results for the first session (the results for the other three tests are similar). We see that already with 50 participants, the standard error is reduced to an acceptable level of 0.05. As we obtained data from more than 140 participants for each test, the obtained gap-level error rates are very reliable.

Range of error rates In our data, the error rates range from 0.01 to 0.99 and are almost continuously distributed. Figure 3 shows an example for the high variance of the gap difficulty within a single paragraph. The error rates in the example are indicated by the size of the circles.

Answer variety Even for the difficult gaps, the students always tried to provide a solution[5] because false answers did not have a negative effect on the result. This behavior leads to a high answer variety (19 different answers per gap on average). The number of provided answers correlates with the error rate (Pearson correlation of 0.57). This indicates that harder gaps trigger more alternatives and do not provoke the same mistake by everyone.

Spelling errors Many of the false answers are variants of the correct solution. The students recognize the solution word but fail to produce it correctly. Unfortunately, the line between a spelling error and a wrong solution cannot be clearly drawn. If a plural *s* is missing we cannot distinguish between a typo and lack of grammatical understanding. Spelling errors often also form new words e.g. *of* vs. *off* or *then* vs. *than* and we cannot decide whether it is a spelling error or a wrong word choice. As the generous time limit allows the students to revise their solutions for typos, we consider them as normal errors in line with Raatz and Klein-Braley (2002).

4 C-Test Difficulty Model

Natural languages are complex and constantly developing constructs that include many exceptions to the rules. Hence, the potential problems for foreign language learners are manifold and hard to anticipate. We took a closer look at the false answers in order to gain deeper understanding of the dimensions that lead to wrong answers and therefore to higher difficulty. We find that the difficulty of C-tests is determined by a combination of many factors.

In order to establish a shared terminology, learner strategies for C-test solving have been categorized as micro-level and macro-level processing strategies (Babaii and Ansary, 2001). Psycholinguistic analyses (Sigott, 2006; Grotjahn and Stemmer, 2002) discuss in detail that both strategies are required for successful C-test solving. Therefore, we developed a model for C-test difficulty that incorporates features from both processing levels (see Figure 4).

Micro-level processing only deals with the solution of the gap and its surrounding micro context. The micro context consists of the word preceding the solution, the solution, and the following word. Both, the preceding and the following word are intact (i.e. not mutilated as gap) and can be used as solution hints by every learner, independent of the performance on the other gaps. In order to determine the difficulty of a gap based on micro-level cues, we estimate two dimensions: the solution difficulty and the candidate ambiguity.

Macro-level processing takes the wider context into account and evaluates the gap in relation to other elements in the sentence and in the whole paragraph. The difficulty of a gap on the macro-level is determined by two dimensions: the inter-gap dependency and the paragraph difficulty.

In the remainder of this section, we elaborate on the individual dimensions. We provide examples that illustrate the described phenomena and introduce the features that operationalize them.

4.1 Solution Difficulty

This micro-level dimension comprises features that approximate whether a learner knows the solution and can correctly produce it in the context. We identified four important phenomena that contribute to the solution difficulty: word familiarity, cognateness, inflection, and phonetic complexity.

[4]For each size, we calculate the error rate based on three randomly selected samples of participants and report the average result.

[5]Except for the weakest students who were not able to understand the texts and left entire paragraphs empty.

Word familiarity If we compare the solutions of the easiest (example 1) and the most difficult gap (example 2), it is obvious that *you* is easier because it is more familiar to the participants than *plentiful*.[6]

1. *If **y**___ are looking for new experiences, ... [you]*
2. *..., people may try self-employment because the opportunities seem **plen**___ and financing is easy to get. [plentiful]*

The probability that a learner knows a word is usually estimated by the word frequency; more frequent words are more likely to be known. We therefore calculate the frequency of the solution and also its length as more frequent words tend to be shorter in English. In previous work, Brown (1989) calculates the frequency of the target word on the basis of the current test text. This is clearly a biased estimate of the frequency, but it is still identified as a good indicator for cloze gap difficulty. Sigott (1995) calculates the frequency of the solution word using counts from the SUSANNE corpus.[7] For our calculations, we use the larger Web1T corpus (Brants and Franz, 2006) and extract normalized probabilities instead of absolute frequencies for better comparison.

Furthermore, a gap is easier to solve, if the solution occurs in a very typical context, e.g. in the micro context *States o_ America*, the candidate *of* is clearly favored, while in the context *write o_ paper*, the candidates *on*, *our* and *off* are more probable. In order to account for typical phrases, we calculate the normalized trigram probability of the micro context.

Even if a word seems familiar to a learner, it might be problematic when used in a compound (e.g. *coastline*) because the prefix only provides information about the first part of the word. In our approach, compounds are detected using a word splitting algorithm with an English dictionary.[8]

Another issue are polysemous words, as learners might know one sense of a word but not be aware of the existence of a second sense. Polysemy interferes with frequency, e.g. the word *well* has a high frequency, but it occurs only rarely in its sense *fountain*. In order to account for polysemy, we count the number of represented word senses for the solution in the lexical-semantic resource UBY (Gurevych et al., 2012).

The two senses of *well* also differ in their word class. The word class has been studied as a difficulty indicator by several researchers but with mixed results. Brown (1989) finds that function words are easier to solve, while Klein-Braley (1996) claims that prepositions are often harder for learners. Sigott (1995) could not confirm any effect of the word class on C-test difficulty. The word class is determined by identifying the part-of-speech (POS) tag. As additional feature, we calculate the probability of the POS sequence of the micro context.

Cognateness Frequency is not the only indicator for word familiarity and can sometimes even be misleading (Beinborn et al., 2014). Many solution words are cognates, i.e. they are very similar to words in other languages like *information* or *laboratory*. In reading comprehension, cognates are known as facilitators because their meaning can be deducted from the form similarity to a word in the mother tongue. We therefore assumed that cognate gaps are easier to solve. However, we observe that they are more likely to trigger production problems. In the 20 gaps with the highest answer variety (33 or more different answers), all solutions have a Latin stem.[9] The 20 gaps with the lowest answer variety (5 or less different answers) are very basic vocabulary.[10]

The production problems are related to the different character combinations and the lower frequency of words with Latin stem. In addition, these words might not be part of the students active vocabulary and are only guessed because they occur as cognates in the students L1. This is supported by the fact that many of the cognate answers resemble orthographic principles from other languages, e.g. for *skeletons* we find **skellets, *skelleton(s), *skelets, *skelletts, *skeletton(s), *skeltons, *skeletes*, and **skelette(s)*.[11]

In order to account for this phenomenon, we estimate the cognateness of words by gathering data from four different lists. We retrieve cognates from UBY using string similarity and from a cognate production algorithm (Beinborn et al., 2013). In addition, we consult the COCA list of academic words[12] and a list of words with latin roots.[13]

Inflection Many errors are caused by wrong morphological inflection as in this example:

*And in **har**___ times like these, ... [harder]*

The base form *hard* (72) is provided more often than the correct comparative *harder* (48), although it is too short. Other inflection errors are caused by singular/plural and adjective/adverb confusion.

In order to account for this phenomenon, we test whether the solution is in lemma form or carries any inflection markers using a lemmatizer. We also check whether the word occurs elsewhere in the text in full form

[6] In all examples, we only highlight a single gap to illustrate a certain phenomenon.

[7] http://www.grsampson.net/RSue.html

[8] http://www.danielnaber.de/jwordsplitter/index_en.html

[9] *appropriate, skeletons, tempting, extract, ancient, private, design, concentrations, state-of-the-art, scientists, modern, examined, constant, essential, stable, entering, basis, synthetic, cost, demands*

[10] *longer, coffee, coffee, in, water, very, give, you, for, people, living, other, number, water, water, from, over, you, over*

[11] DE: *Skelett*, FR: *squelette*, ES: *esqueleto*, NL: *skelet*

[12] http://www.academicvocabulary.info/

[13] http://en.wikipedia.org/wiki/List_of_Latin_words_with_English_derivatives

(i.e. not as a gap) because it facilitates the correct production for the student. This feature is comparable to the semantic cache used by Brown (1989).

Phonetic complexity Wrong answer variants for C-test gaps are often rooted in phonetic problems. The spelling of a word is more difficult, if it contains a rare sequence of characters. The word *appropriate*, for example, triggers 69 different answers, 40 of them were provided only once. In addition, a spelling error is more likely to occur, in words with rare grapheme-phoneme mapping as in *Wednesday*. We build a character-based language model that indicates the probability of a character sequence using BerkeleyLM (Pauls and Klein, 2011). In addition, we build a phonetic model using phonetisaurus, a statistical alignment algorithm that maps characters onto phonemes.[14] Both models are trained only on words from the Basic English list in order to reflect the knowledge of a language learner.[15] Based on this scarce data, the phonetic model only learns the most frequent character-to-phoneme mappings and assigns higher phonetic scores to less general letter sequences. We use this score as a feature and additionally calculate the string similarity between the output and the correct pronunciation in the CMU dictionary.[16]

Another source for phonetic problems occurs, if the prefix boundary splits the word in a way that leads to another pronunciation pattern compared to the solution word as in this example.

It is not easy to design and build a ***mac***___ *that is both, efficient and durable. [machine]*

Due to the syllable split, the prefix provokes answers with the pronunciation [mac] such as *macanics*, *mac(h)anism*, *macanical*, *macbook*, *macphone*, and *macro* instead of the original pattern [maʃ]. A similar issue occurs when the prefix splits a compound such as *greenhouse*. We check if the prefix boundary occurs within a compound or a syllable using a hyphenation dictionary.[17]

4.2 Candidate Ambiguity

This micro-level dimension examines whether a competing candidate is more accessible for the learner in the given context. Even if the learner is familiar with the solution word, she might still not be able to produce it, because a competing candidate is stronger. For example, in 42 gaps in our data, an alternative answer is provided more frequently than the intended solution.

Some of these gaps actually have more than one possible solution as the following example:

Scientists have ***ma***___ *tools that allow them to extract subtle information from ancient bones and their environmental settings. [many]*

Instead of the correct solution *many* (89), most students provided *made* (238) which can also be considered correct here. These cases had not been anticipated by the language teachers, they only encoded one solution in the system.

In other cases, alternative answers seem very probable to the students but are nevertheless false.

A natural blanket of greenhouse gases in the atmosphere keeps the planet warm enough for life as we know it at a comfortable 15C today. Human-caused emissions of greenhouse gases have made the blanket ***thi***___*, trapping heat and leading to a global warming. [thicker]*

Instead of the correct solution *thicker* (12), the students provided many alternative solutions more often: *thinner* (31), *thin* (19), *thick* (18), *this* (14), *thing* (14). *Thinner* fits syntactically but completely changes the semantics of the sentences as it is the antonym of the correct solution. The learner needs to apply world knowledge to understand that a thinner blanket would not trap heat. In our model, we want to account for both cases, as it would be very helpful if ambiguous gaps could be automatically detected. This aspect has been neglected in previous work on C-test difficulty.

In order to account for competing candidates, we first determine the candidate space and then describe our features approximating the probability that a competing candidate confuses the student.

Candidate space Prior to the tests, the students are informed about the quite restrictive length constraint. The given prefix of C-test gaps consists always of the smaller half of the solution: if 3 characters are provided as prefix, the correct word can only consist of 6 or 7 characters. This can be a useful indicator for the solution, but the data reveals that in approx. 40% of the false answers, the length constraint is not respected. The absolute number of false length answers is higher for weaker students. However, the proportion of false length answers relative to all false answers is higher for stronger (0.45) than for weaker (0.32) students.

Length violations can be caused by candidates that seem viable for the context and are more accessible than the solution or by wrong inflection of the word ending. In other cases, it is obvious that the student does not find a proper solution and provides just anything that remotely fits. It would be interesting to repeat the test with the constraint that false answers have a negative influence on the overall score in order to find out whether the students are aware of the length violation.

Bresnihan and Ray (1992) show that students perform

[14] http://code.google.com/p/phonetisaurus/

[15] http://ogden.basic-english.org

[16] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[17] http://hindson.com.au/info/free/free-english-language-hyphenation-dictionary/

better on the C-test, if the length of the solution is graphically indicated by dashes or dots which supports the assumption that length violations are often not noticed in the standard C-test. As we want to account for this phenomenon, we decide to relax the length constraint. We only allow a length tolerance of 1, i.e. for a prefix of length 3, we consider candidates with 5 to 8 (instead of 6 to 7) characters, as the candidate space would be too large otherwise.

We noticed that even candidates with wrongly spelled prefix can be competitors, e.g. some students provided the answer **demage* for the prefix *dem* instead of the correct solution *demands*. The word *damage* actually fits semantically into the gap, but as the prefix is different, we currently do not add such cases to the candidate space.

In order to account for candidate ambiguity, we rank all candidates according to three criteria: the unigram frequency, the trigram frequency of the micro context, and the parse score. Statistical parsers usually provide parse scores in order to determine the best variant. This score cannot be used as an absolute value because it depends on the sentence length but it helps to distinguish between candidates. A candidate that produces another parse tree than the solution is less likely to be correct. For each ranking, we determine the rank of the solution and the number of candidates above a fixed threshold.

In addition, we take the intersection of the best candidates from the above rankings, combine them into a set of top candidates that are likely to compete with the solution and determine its size. Moreover, we calculate the maximum string similarity of the candidates with the solution in order to capture very close variants (e.g. *base* and *basis*).

4.3 Inter-gap dependency

This macro-level dimension assesses the dependency of the current gap on previous gaps: can it be solved, even if the previous gap has not been solved? In previous work, Harsch and Hartig (2010) examine dependencies between individual gaps using a Rasch testlet model and find that some gaps strongly depend on each other, while others can be solved independently.

At the same time, fertility is set to fall as women leave childbirth ***la***___ *and* ***la***___. *[later]*

In these gaps, *later* is repeated which makes it easy to fill in the second gap, if the first one is solved.

The dependency of a gap is related to its position and the difficulty of the preceding word. If a gap is preceded by a very difficult gap, the available context is damaged which can have an effect on the difficulty of the following gaps. A gap occurring towards the end of a sentence, is also more likely to be influenced by limited context. We thus calculate the position of the gap and the number of previous gaps in the sentence and in the paragraph. We check if the same solution also occurs in another gap to account for repetition. In order to estimate the difficulty of the previous gap, we calculate its unigram and trigram probability. If we already had a good difficulty prediction algorithm, we could perform incremental prediction and use the difficulty label of the previous gap as a feature for the current gap, but this is left to future work.

In addition, we check for gaps with the prefix *th* because they enable many reference words such as *this*, *that*, *there*, *then*, *these*, *those*, *they*, and *their*. The student needs to perform co-reference resolution in order to select the correct word. These referential gaps usually cannot be solved on the basis of the micro context.

4.4 Paragraph difficulty

This macro-level dimension determines whether the learner is generally able to understand the text. The overall difficulty of a paragraph contributes to the difficulty of the individual gaps because more complex texts are harder to parse for language learners, especially when every second word is a gap. Thus, the available context for each gap is assumed to be lower in more difficult paragraphs. As we have seen in Section 3.1, the difficulty of the gaps within one paragraph varies strongly. We therefore assume that the paragraph difficulty only adds a constant effect to the overall gap difficulty.

The difficulty of a paragraph is inversely related to its readability. We calculate the following readability features for the whole paragraph and for the sentence containing the gap. Average word and sentence length are the underlying basis of traditional readability measures such as Flesch-Kincaid and Fry which correlate with cloze test difficulty according to Brown (1989). We calculate both, but do not find much variety as the paragraphs in our data are all of comparable length (64-99 words, 3-7 sentences, 4.85 characters per word).

The type-token ratio, the verb variation, and the pronoun ratio are used as indicators for lexical diversity and referentiality. Klein-Braley (1984) already determined the type-token ratio as useful cue for paragraph difficulty prediction. We also use syntactic readability features such as the number of entity mentions, the number of certain POS types (e.g. noun, determiner, adjective) and the number of certain phrase patterns (e.g. verbal phrase, noun phrase, subordinate phrase).

Having introduced all four dimensions of C-test difficulty, we now report on the results of the actual difficulty prediction. Difficulty prediction of C-tests has up to now only been performed on the paragraph level (Klein-Braley, 1984; Traxel and Dresemann, 2010). In this article, we go beyond paragraphs and predict the difficulty of gaps. We first determine the human performance on the task and use it as a reference for the performance of the machine learning approach based on our difficulty model.

	A1	A2	A3	Median A1-A3
Correct Prediction	200	209	192	213
Overestimation	90	99	83	101
Underestimation	107	89	118	84
NA	2	2	6	1
Accuracy	0.50	0.52	0.48	0.53

Table 2: Results of the human annotations

5 Human Difficulty Prediction

Due to the high number of participants, we already have precise gap-level error rates (cf. Figure 2) for our tests. We now want to determine to what extent human annotators are able to predict these error rates. For this purpose, we asked three English language teachers to assign a difficulty category to each gap according to the following scheme:

1: Very easy gap (error rate ≤ 0.25)
2: Easy gap ($0.25 <$ error rate ≤ 0.5)
3: Medium gap ($0.5 <$ error rate ≤ 0.75)
4: Difficult gap (error rate > 0.75)

The annotation was performed on the same 20 texts as described in Section 3.1. The teachers were already familiar with these texts, as they had chosen them for the testing period. We consider a gap to be correctly annotated, if the human-assigned class matches the actual error rate.

Given the highly experienced annotators, the prediction accuracy is lower than expected. The three annotators obtain comparable accuracy, each of them correctly predicts approximately 50% of the gaps (see Table 2). There is no obvious bias in the annotations, difficulty is both under- and overestimated. If we combine the human prediction by taking the median of the three annotators, 53.4% are annotated correctly. These results show that even experienced teachers are not able to foresee all factors that influence the difficulty of a gap.

Somewhat surprisingly, the agreement between the annotators is also low. The Fleiss' Kappa for the three annotators is 0.36, the pairwise comparison ranges from 0.31 to 0.39. Only in 38.6% of the gaps, all three annotators agreed with each other. For only 25.3%, all three annotators agreed with each other *and* with the actual measured error rate. This shows that human difficulty prediction is quite subjective.

The mediocre human performance on the task reveals the complexity of predicting the elements of language that cause problems for foreign language learners. However, this strengthens the need for reliable prediction methods like the one described in this paper. Note that the automatic prediction is compared with the actual error rates, not the human predicated ones. Thus, it is possible to outperform human performance with automatic methods and provide a very helpful tool.

	Classification			Regression	
	P	R	F_1	Pearson's r	RMSE
Majority Baseline	.19	.43	.26	.00	.25
Sigott (1995)	.23	.40	.28	.34	.24
Our Approach	.46	.48	.46	.64	.20
Human Median	.56	.53	.54	-	-

Table 3: Results for leave-one-out crossvalidation on the training set for regression and classification prediction (both trained on support vector machines). Classification results are the weighted average of precision (P), recall (R) and F_1-measure over all four classes.

6 Automatic Difficulty Prediction

Our difficulty prediction approach is based on the model described in the previous section. We extract the features using tools for natural language processing provided by DKPro Core (de Castilho and Gurevych, 2014). We then perform experiments with different datasets and classifiers using Weka (Hall et al., 2009) through the DKPro TC framework (Daxenberger et al., 2014).[18]

6.1 Classification vs Regression

For the human annotation, we used a classification scheme because assigning difficulty scores on a fine-grained numerical scale would be too challenging even for experienced teachers. However, as the actual error rates are continuously distributed, gaps that are close to the class boundaries are more likely to be mislabeled. Therefore we also test regression prediction using the actual error rates instead of the artificially determined classes. We perform leave-one-out testing on the training set in order to determine the best approach.

We compare our model against the human performance and two baselines: A naive one that predicts the majority class for classification and the mean value for regression and one that only uses the features proposed by Sigott (1995) (solution probability, word class of solution, and constituent type of gap).

In Table 3, we report weighted precision, recall and F_1-measure over all classes for classification and Pearson correlation and root mean squared error for regression. It can be seen that our approach clearly outperforms the baselines in both cases.

For classification, the human median annotation is better than our approach. In order to also compare our regression results to the human annotations, we map the numerical predictions back into classes according to the scheme explained in the previous subsection. The quadratic weighted kappa considers the classes on an ordinal scale and thus gives a better impression of the usefulness of the prediction. The results in Table 4 show that

[18] More information on data and resources can be found at http://www.ukp.tu-darmstadt.de/data/c-tests.

	q.w. κ
Human Median	.59
SMO Classification	.47
Mapped SMO Regression	.58

Table 4: Quadratic weighted κ of difficulty class predictions

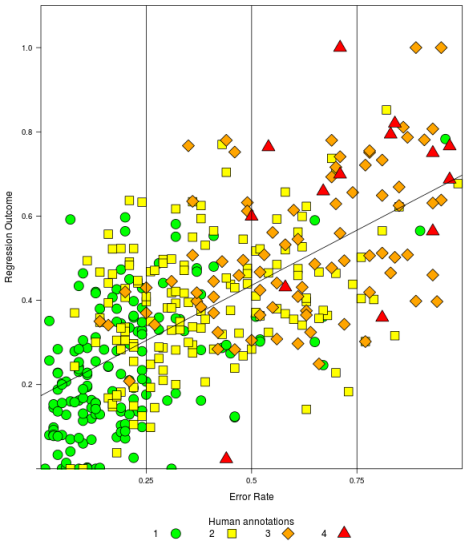

Figure 5: Regression results for leave-one-out testing on the training data. The symbols indicate the difficulty class that was annotated by the human experts.

the performance of the regression approach is almost on the same level as the median of the human prediction. Therefore, we will focus on regression prediction for the remainder of the paper.

For a better understanding of the behaviour of human and automatic predictions, the plot in Figure 5 combines the two results. The position in the plot indicates the relation between the true error rate and the prediction and the symbols show the corresponding human annotation. The plot reveals that the regression equation predicts the right tendency but tends to slightly underestimate difficult gaps and overestimate easy gaps. The human prediction performs well for the easiest gaps (class 1, green circle) while the other three classes are confused quite often.

6.2 Feature selection

For a deeper analysis of our difficulty model, we now compare different feature groups.

Processing Levels The results in the first two rows show that the gap difficulty is mainly determined by the features representing micro-level processing. This is not surprising, as these features are calculated for each gap, while most of the macro-level features are constant for all gaps in the paragraph. The predictive power on the micro-level of our approach is a strong improvement over previous prediction approaches that only attempted to predict paragraph difficulty.

	Feature Group	# Feat.	Pearson's r
Micro vs. Macro	Micro-level Processing	51	.50**
	Macro-level Processing	37	.24**
Dimensions	w/o Solution Difficulty	50	.42**
	w/o Candidate Ambiguity	74	.54**
	w/o Inter-Gap Dependency	79	.62*
	w/o Paragraph Difficulty	59	.59**
All		87	.64

Table 5: Regression results for different feature groups. Significant differences to the result with all features are indicated with *($p<0.05$) and **($p<0.01$).

Dimensions The middle part of Table 5 shows that the prediction results decrease significantly, if we exclude features from one dimension. The effect is particularly strong, if we exclude the features estimating the difficulty of the solution, while the effect of the inter-item dependency features is quite small. This supports previous theoretical research claiming that the solution word itself and its micro context are most relevant for the solving processes. The dimensions candidate ambiguity and inter-item dependency have been newly introduced, while many of the features for solution and paragraph difficulty are well established. We therefore assume that future work on improved feature development for these dimensions could lead to even better prediction results.

Selected Features As the results for the individual dimensions might be related to the number of features, we additionally perform feature selection and reduce the set to 21 features.[19]

The selection shows that the probability of the word, the phrase and the character sequence play a major role for prediction. However, it might be the case that the continuous values of these features are simply more suitable for regression approaches than boolean features such as the word class of the solution. In addition, the number of available candidates plays an important role but priming effects also need to be considered (whether the solution occurs previously in the text or mutilated as another gap). For the paragraph difficulty, the number of verbs and embedded sentences seems to be a good indicator of difficulty.[20]

Interestingly, features from all four dimensions are included in the selection as can be seen in Table 7. This indicates that the dimensions in our model represent the factors that have an influence on the C-test difficulty quite

[19] We use the WrapperSubsetEval-evaluator with SMOreg and BestFirst-search as implemented in Weka.

[20] The term *CoverSentence* in Table 6 refers to the sentence containing the gap.

Dimensions	Selected Features
SolutionDifficulty	IsAdverb IsPlural CharacterLMProbabilityOfPrefix UnigramProbability LeftBigramProbability RightBigramProbability TrigramProbability SolutionOccursAsText
CandidateAmbiguity	NrOfCandidates NrOfParseCandidates RankOfSolutionInParseCandidates MaxLCSRofCandidatesAndSolution
Inter-Item Dependency	SolutionOccursInAnotherGap PrefixIsTh NumberOfPreviousGapsInCoverSentence PositionOfGap
Paragraph Difficulty	AvgWordLength NounsPerSentence VerbsPerSentence VerbVariation SBarsInCoverSentence

Table 6: Selected Features

	All Features	Selected Features
Solution Difficulty	37 (43%)	8 (38%)
Candidate Ambiguity	13 (15%)	4 (19%)
Inter-Gap Dependency	8 (9%)	4 (19%)
Paragraph Difficulty	28 (33%)	5 (24%)
Sum	87	21

Table 7: Proportion of dimensions in selected features and all features

well. However, the solution difficulty dimension is by far the most important one, while the other three dimensions contribute fewer features. We include the prediction results for the selected features in Table 8 which is discussed in the following section.

6.3 Test results

In order to evaluate our model on unseen data, we test it on a set of 375 additional gaps. The results on the test set are substantially worse than in the leave-one-out (LOOCV Train) setting. If we merge the two sets and perform leave-one-out testing on the whole data (LOOCV All), the results get close to our training set again. This indicates that the test set contains characteristics, that have not been observed during training. It is also interesting that using only the selected features yields better results on the smaller training set, while the full model is better on larger data. In order to support the assumption that our model performs better with more data, we plot a learning curve (see Figure 6). We calculated the Pearson correlation for increasing sample sizes of randomly selected instances and average the results over 100 runs. The anomaly for smaller sample sizes can be explained by very high standard deviations. Starting from a sample size of about 70 instances, the learning curve proceeds as

	#	LOOCV Train	Train-Test	LOOCV All
Mean Baseline	1	.00	.00	.00
Sigott (1995)	7	.34	.38	.36
Full Model	87	.64	.32	.60
Selected Features	21	.68	.44	.57

Table 8: Results on the train and the test set

expected and highlights the importance of a larger training set.

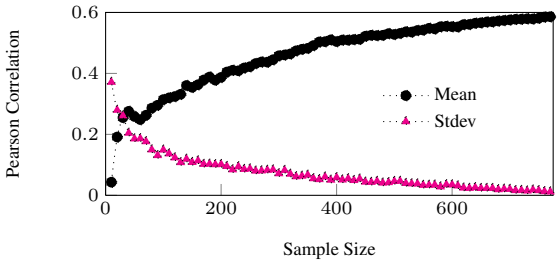

Figure 6: Learning curve for 10-fold cross-validation with increasing size of training data, results are averaged over 100 runs

6.4 Error Analysis

Figure 7 shows that our prediction approach produces a few strong outliers for the test data. In particular, it strongly underestimates the error rate for some very easy gaps. We perform an error analysis on the 9 outliers.

Underestimation In two underestimated gaps, the solution requires an apostrophe (*Earth's*, *world's*). This has not been seen in the training data, and therefore we cannot predict that the students have difficulties here. It is debatable whether punctuation should be included into the solution but the language teachers insisted on the importance of these gaps. In two other cases, the students systematically favour a wrong solution—one is due to

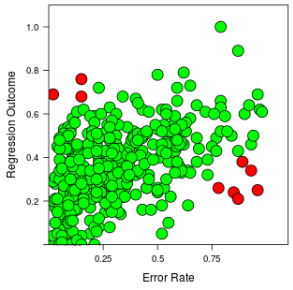

Figure 7: The prediction for the train-test setting produces more outliers. Instances with an absolute difference of predicted and actual error rate ≥ 0.5 are coloured red.

spelling (*of* instead of *off*) and the other due to referentiality (*the* instead of *this*)—which our approach did not anticipate correctly. The last two outliers occur in a phrase that is very frequent for native speakers but nevertheless unknown to the participants (*cause untold damage*, *the continental United States*[21]).

Overestimation One of the items for which the error rate is strongly underestimated is the compound *carbon-free*. It can be seen, that the teachers deviated from the original length constraint here and applied it only on the second part of the component. As these kind of compounds have not been seen in the training set, our approach estimates the difficulty for providing *carbon-free*, while it should rather consider only *free*. The second overestimation is due to the named entity *Deutsche Bahn*, which is unlikely to occur in English text but very common for students living in Germany. The third overestimated outlier is simply due to an unfortunate combination of a long word (*dangerous*) at the end of a difficult sentence that is nevertheless easy for the students.

The errors due to apostrophes and hyphenated compounds can be minimized by adapting the processing. In order to also anticipate the other outlier phenomena, we need more training data.

7 Conclusions

We introduce the first model for the automatic prediction of gap-level difficulty of C-tests. We collected data from real learners and find that the gap-level error rates are quite stable. The prediction results of our approach are on the same level as the performance of human experts. The learning curve indicates that even better results are possible with more training data. A higher number of instances makes it easier to learn the nuances for the prediction and this can help to improve the features.

Our work also sheds light on the question what C-tests measure. The difficulty of a C-test gap is determined by a combination of many factors. Our experiments have shown that both, micro- and macro-level cues, contribute to the gap difficulty: i) problems related to the solution such as spelling, phonetic difficulties and morphological derivation, ii) problems caused by competing candidates, iii) problems caused by dependencies between gaps, and iv) readability problems caused by text complexity. Even the reduced set of selected features comprises features from all introduced dimensions which shows that our conclusions drawn from the data analysis led to a very suitable model. However, the features measuring the difficulty of the solution and the probability of the micro context seem most relevant. As a next step, we need to improve the feature extraction for compound nouns and named entities.

[21]The students provided only *continent* instead.

Our approach has already raised interest in language teachers who see strong practical benefits. The automatic difficulty prediction facilitates test selection, as teachers can run our approach on a corpus and only inspect tests with adequate difficulty. The system could also be tuned towards the prediction of potentially ambiguous gaps so that teachers become aware of the alternative solutions. In addition, our approach can also be used productively for the automatic test generation in platforms for self-directed language learning.

Our model has been developed for the difficulty prediction of English C-tests. However, it can also be generalized to other languages and to test variants of reduced redundancy testing. In future work, we aim at adapting the difficulty of a given text by varying the gap placement.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Klaus Tschira Foundation under project No. 00.133.2008. We thank the anonymous reviewers for their very helpful comments.

References

Manish Agarwal and Prashanth Mannem. 2011. Automatic Gap-fill Question Generation from Text Books. *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64.

Esmat Babaii and Hasan Ansary. 2001. The C-test: a valid operationalization of reduced redundancy principle? *System*, 29(2):209–219.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2013. Cognate Production using Character-based Machine Translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 883–891. Asian Federation of Natural Language Processing.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Readability for foreign language learning: The importance of cognates. *International Journal of Applied Linguistics*, pages 136–162.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1.1. *Linguistic Data Consortium*.

Brian Bresnihan and Stratton Ray. 1992. C-tests and the usefulness of non-linguistic instructions. In Rüdiger Grotjahn, editor, *Der C-Test. Theoretische Grundlagen und praktische Anwendungen 1*, pages 193–216. Brockmeyer, Bochum.

Jonathan C Brown, Gwen A Frishkoff, and Maxine Eskenazi. 2005. Automatic Question Generation for Vocabulary Assessment. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 819–826, Morristown, NJ, USA. Association for Computational Linguistics.

James Dean Brown. 1989. Cloze item difficulty. *JALT journal*, 11:46–67.

C. A. Chapelle. 1994. Are C-tests valid measures for L2 vocabulary research? *Second Language Research*, 10(2):157–187, June.

Andrew D. Cohen. 1984. The C-Test in Hebrew. *Language Testing*, 1(2):221–225.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch. 2014. DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66. Association for Computational Linguistics.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014*, pages 1–11, August.

Thomas Eckes and Rüdiger Grotjahn. 2006. A closer look at the construct validity of C-tests. *Language Testing*, 23(3):290–325, July.

Thomas Eckes. 2011. Item banking for C-tests: A polytomous Rasch modeling approach. *Psychological Test and Assessment Modeling*, 53(4):414–439.

Rüdiger Grotjahn and Brigitte Stemmer. 2002. C-Tests and language processing. In James A. Coleman, Rüdiger Grotjahn, and Ulrich Raatz, editors, *University language testing and the C-Test*, pages 115–130. AKS-Verlag, Bochum.

Rüdiger Grotjahn, Christine Klein-Braley, and Ulrich Raatz. 2002. C-Tests: an overview. In James A. Coleman, Rüdiger Grotjahn, and Ulrich Raatz, editors, *University language testing and the C-Test*, pages 93–114. AKS-Verlag, Bochum.

Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M Meyer, and Christian Wirth. 2012. A Large-Scale Unified Lexical-Semantic Resource Based on LMF. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 580–590.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. 11(1).

Claudia Harsch and Johannes Hartig. 2010. Empirische und inhaltliche Analyse lokaler Abhängigkeiten im C-Test. In Rüdiger Grotjahn, editor, *Der C-Test: Beiträge aus der aktuellen Forschung*, pages 193–204. Peter Lang.

Michael J. Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Proceedings of NAACL-HLT*, pages 460–467.

A. Jafarpur. 1995. Is C-testing superior to cloze? *Language Testing*, 12(2):194–216, July.

Gerhard Jakschik, Hella Klemmert, and Dorothea Klinck. 2010. Computergestützter Multiple Choice C-Test in der Bundesagentur für Arbeit: Bundesweite Erprobung und Einführung. In Rüdiger Grotjahn, editor, *Der C-Test: Beiträge aus der aktuellen Forschung The C-Test: Contributions from Current Research*, pages 231–264. Peter Lang International Academic Publishers.

Tadamitsu Kamimoto. 1993. Tailoring the Test to Fit the Students: Improvement of the C-Test through Classical Item Analysis. *Language Laboratory*, 30:47–61, November.

Christine Klein-Braley and Ulrich Raatz. 1984. A survey of research on the C-Test. *Language Testing*, 1(2):134–146, December.

Christine Klein-Braley. 1984. Advance Prediction of Difficulty with C-Tests. In Terry Culhane, Christine Klein-Braley, and Douglas K. Stevenson, editors, *Practice and problems in language testing*, volume 7, pages 97–112.

Christine Klein-Braley. 1985. A cloze-up on the C-Test: a study in the construct validation of authentic tests. *Language Testing*, 2(1):76–104.

Christine Klein-Braley. 1996. Towards a theory of C-Test processing. In Rüdiger Grotjahn, editor, *Der C-Test. Theoretische Grundlagen und praktische Anwendungen 3*, pages 23–94. Brockmeyer, Bochum.

Christine Klein-Braley. 1997. C-Tests in the context of reduced redundancy testing: an appraisal. *Language Testing*, 14(1):47–84, March.

Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2):177–194, May.

Jack Mostow and Hyeju Jang. 2012. Generating Diagnostic Multiple Choice Comprehension Cloze Questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146. Association for Computational Linguistics.

Adam Pauls and Dan Klein. 2011. Faster and Smaller N-Gram Language Models. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, volume 1, pages 258–267. Association for Computational Linguistics.

Laura Perez-Beltrachini, Claire Gardent, and German Kruszewski. 2012. Generating Grammar Exercises. pages 147–156.

Ulrich Raatz and Christine Klein-Braley. 2002. Introduction to language testing and to C-Tests. *University language testing and the C-test*, pages 75–91.

Keisuke Sakaguchi, Yuki Arase, and Mamoru Komachi. 2013. Discriminative Approach to Fill-in-the-Blank Quiz Generation for Language Learners.

Günther Sigott. 1995. The C-test: some factors of difficulty. *AAA. Arbeiten aus Anglistik und Amerikanistik*, 20(1):43–54.

Günther Sigott. 2006. How fluid is the C-Test construct. In Rüdiger Grotjahn and Günther Sigott, editors, *Der C-Test: Theorie, Empirie, Anwendungen The C-Test: Theory, Empirical Research, Applications*, pages 139–146. Peter Lang.

David Singleton and David Little. 1991. The second language lexicon: some evidence from university-level learners of French and German. *Second Language Research*, 7:61–81.

Adam Skory and Maxine Eskenazi. 2010. Predicting Cloze Task Quality for Vocabulary Training. In *The 5th Workshop on Innovative Use of NLP for Building Educational Applications (NAACL-HLT)*.

Bernard Spolsky. 1969. Reduced Redundancy as a Language Testing Tool. In G.E. Perren and J.L.M. Trim, editors, *Applications of linguistics*, pages 383–390. Cambridge University Press, Cambridge, August.

Oliver Traxel and Bettina Dresemann. 2010. Collect, callibrate, compare: A practical approach to estimating the difficulty

of C-Test items. In Rüdiger Grotjahn, editor, *Der C-Test: Beiträge aus der aktuellen Forschung The C-Test: Contributions from Current Research*, pages 57–69. Peter Lang International Academic Publishers, Frankfurt a.M.

Lev Vygotsky. 1978. Mind in society: The development of higher psychological processes. Harvard University Press.

Torsten Zesch and Oren Melamud. 2014. Automatic generation of challenging distractors using context-sensitive inference rules. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148. Association for Computational Linguistics.