Candidate Evaluation Strategies for Improved Difficulty Prediction of Language Tests

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

◇ UKP Lab, Technische Universität Darmstadt

‡ Language Technology Lab, University of Duisburg-Essen

§ UKP Lab, German Institute for Educational Research

http://www.ukp.tu-darmstadt.de

Abstract

Language proficiency tests are a useful tool for evaluating learner progress, if the test difficulty fits the level of the learner. In this work, we describe a generalized framework for test difficulty prediction that is applicable to several languages and test types. In addition, we develop two ranking strategies for candidate evaluation inspired by automatic solving methods based on language model probability and semantic relatedness. These ranking strategies lead to significant improvements for the difficulty prediction of cloze tests.

1 Introduction

In learning scenarios, evaluating the learner's proficiency is crucial to assess differences in learner groups and also individual learner progress. This kind of evaluation is usually performed over the learner's results on certain tasks or tests. For informative results, it is important that the test difficulty is suitable for the learner. It needs to be challenging enough to avoid boredom and stagnation, but the learner should still be able to solve the task at least partially. In this work, we focus on language proficiency tests and aim at predicting the difficulty for five different test datasets.

Understanding the challenging elements of a task is an essential prerequisite for learner support. In natural language processing, human performance is usually considered as the gold standard for automatic approaches. The models are tuned and adjusted to reach human-like results. In learning settings, the human performance is flawed because of limited knowledge and lack of experience. In this work, we thus apply a reverse approach: we exploit strategies from automatic solving to model human difficulties.

To enable the experiments, we retrieved datasets from various testing institutions and conducted a learner study to obtain error rates for an additional test type.[1] For a better understanding of the differences between test types, we first calculate the candidate space of potential answers and compare it to learner answers. We assume that higher answer ambiguity leads to higher difficulty. As all datasets allow binary scoring (correct/wrong), the difficulty of an item is interpreted as the proportion of wrong answers, also referred to as the error rate. We then build a generalized difficulty prediction framework based on an earlier approach we presented in Beinborn et al. (2014a) which was limited to English and to one specific test type. We evaluate the prediction for different test types and languages and obtain remarkable results for French and German.

Many language tests are designed as multiple choice questions. The generalized prediction approach lacks predictive power for this format because the evaluation strategy for the answer candidates is solely based on word frequency. We develop two strategies for more sophisticated candidate ranking that are inspired by automatic solving methods based on language models and semantic relatedness. We show that the candidate ranking can successfully model human evaluation strategies and leads to improved difficulty prediction for cloze tests.

[1] The dataset is available at: https://www.ukp.tu-darmstadt.de/data/c-tests

In order to establish common ground, we first introduce the concept of reduced redundancy testing and the most popular test types.

2 Reduced Redundancy Tests

In language learning, most proficiency tests rely on the principle of reduced redundancy testing as introduced by Spolsky (1969). He formalized the idea that "natural language is redundant" and that the proficiency level of language learners can be estimated by their ability to deal with reduced redundancy. For testing, redundancy can be reduced by eliminating (partial) words from a text to create a gap. The learner is then asked to fill in the gaps i.e. to complete the missing words.

Reduced redundancy tests can be distinguished into *open* and *closed* answer formats. In open formats, the learner has to actually produce the solution, while it can be selected from a small fixed set of multiple choice options in closed formats. This technique provides full control over the candidate space, but the selection of good answer options (distractors), that are not a proper solution, is a difficult task. Most previous works in the field of educational natural language processing focus on the generation of distractors to manipulate the difficulty, i.e. for cloze tests (Zesch and Melamud, 2014; 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).

In addition to the answer format, the test types can be distinguished by the gap type and the deletion rate. On the local level, the gap type determines which portion of the word is deleted. On the global test level, the deletion rate determines the distribution of gaps in the text. A higher number of gaps per sentence results in a higher redundancy reduction. This increases the dependency between gaps as the mutilated context of a single gap can only be recreated by solving the surrounding gaps.

2.1 Cloze test

Cloze tests have been introduced by Taylor (1953) and have become the most popular form of reduced redundancy testing. In cloze tests, full words are deleted from a text. This strategy requires comprehensive context, so the deletion rate is usually every 7th word or higher (Brown, 1989). The main problem with cloze tests is that the gaps are usually highly ambiguous and the set of potential solutions cannot be exactly anticipated (Horsmann and Zesch, 2014). Therefore, most cloze tests are designed as closed formats, so that the correct solution can be selected from a set of distractors (see Figure 1 for an example).

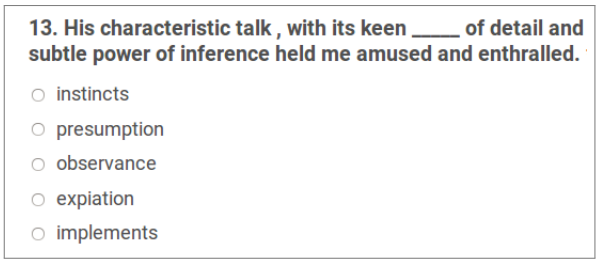

Figure 1: Example for a cloze question, the solution is *observance*.

2.2 C-test

Although the cloze test is widely used, the setup contains several weaknesses such as the small number of gaps and the ambiguity of the solution. The C-test is an alternative of the cloze test that has been developed by Klein-Braley and Raatz (1982). The C-test construction principle enables a higher number of gaps on less text, every second word of a short paragraph is transformed into a gap. As this high deletion rate would lead to an unfeasible degree of redundancy reduction, only the second "half" of the word is deleted to narrow down the candidate space, see the example below.

Vacc___ like penic___ and ot___ antibiotics th___ were disco___ as a dir___ result are lik___ the grea___ inventions o___ medical sci___.[2]

2.3 Prefix deletion test

The prefix deletion test is a more difficult variant of the C-test that can be used to assess more advanced students up to native speakers (Sigott and Köberl, 1996). In this case, the first "half" of the word (the prefix) is deleted. As word endings vary less than word onsets (at least for the languages under study), the candidate space is increased and allows alternative solutions that are equally valid. See the previous

[2]Solutions: *Vaccines, penicillin, other, that, discovered, direct, likely, greatest, of, science*

example as a prefix deletion test below.

___ines like ___illin and ___er antibiotics ___at were ___vered as a ___ect result are ___ely the ___test inventions ___f medical ___nce.

In standard C-tests, a big challenge is to select the correct inflection of the solution, especially for languages with a rich morphology. In prefix deletion tests, the inflected ending of the word is already provided and thus the focus is shifted towards semantic challenges. Psycholinguistic experiments have shown that the information value of the initial part of a word is higher than the final part (Broerse and Zwaan, 1966; Kinoshita, 2000). This supports the assumption that prefix deletion tests are more difficult.

In general, the following hypothesis is supposed: A higher degree of redundancy reduction for the gap results in a bigger candidate space and leads to increased difficulty (compare the results by Sigott and Köberl (1996)). In the following section, we provide an approximation of the candidate space for each test variant.

3 Candidate Space

The main difference between the different test types is the number of competing candidates. In this section, we analyze the candidate space for the three languages English, French and German and for the test types cloze, C-test and prefix deletion. We calculate the candidates for each word in the vocabulary and then average the results for words with the same length to approximate the candidate space.

Language	Words	Mean word length
English (American)	99,171	8.5 ±2.6
French	139,719	9.6 ±2.6
German	332,263	12.0 ±3.5

Table 1: Vocabulary size and mean word length for different languages

Candidate space for different languages We focus on English, French and German because they are used in our datasets. The word list package provided by Ubuntu for spell-checking serves as vocabulary.[3]

The size of the lists vary depending on the morphological richness of the language; the German list is more than three times bigger than the English one (see Table 1). It should also be noted that the average word length is much higher for German. This is mainly due to the existence of noun compounds that concatenate two or more words into one.

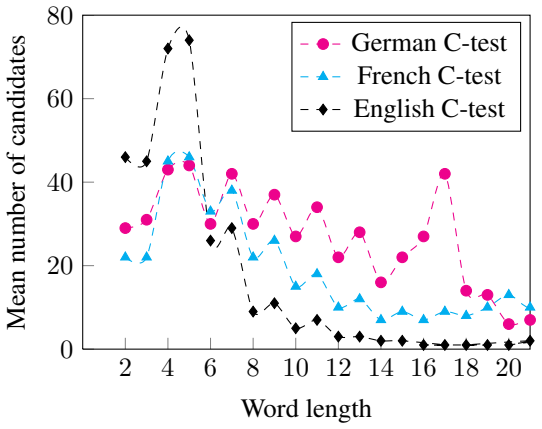

Figure 2: Mean number of candidates for different test types with respect to word length

Figure 2 illustrates how the candidate space varies for the languages under study. It can be seen that for English the candidate space is maximized for extremely short words and decreases rapidly with increased word length. In comparison, the French and in particular the German candidate space is more leveled: it is smaller for short words, but bigger and more constant for longer words.

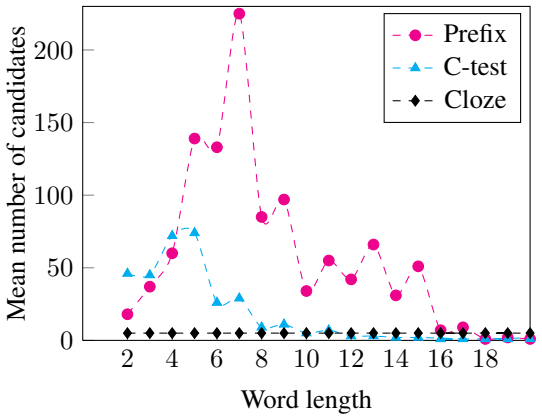

Figure 3: Mean number of English candidates for different test types with respect to word length

Candidate space for different test types Figure 3 shows the English candidate space for the test types.

[3] http://packages.ubuntu.com/de/lucid/wordlist, 15.12.2014

The number of candidates for the cloze test with five distractors is of course always five. Compared to the C-test, the candidate space for the prefix deletion test is extremely large, in particular for words with medium length (five to nine characters). This could be an explanation why this test type is considered to be more difficult than the standard C-test. However, following this hypothesis, the cloze tests should be fairly easy given the consistently small candidate space. The obtained error rates and the feedback of our test participants do not support this assumption. This gives rise to the idea that the candidate space considered by the learner differs from the computational one.

Candidate evaluation by learners When solving open formats, the learners cannot consider the full candidate space; only the words that are in the active vocabulary of the learner are accessible. In addition, the context can lead to priming effects and the test situation might alter the stress level of the participant and apply further restrictions.

From the above arguments, one would expect that the learner's candidate space is smaller than the objective candidate space. However, we need to take into account that learners also consider wrong options, see the different learner answers for the gap *appro___* in Figure 4, for example. The computational candidate space on the left consists of only 9 candidates, but the participants provided 68 different answers along with the solution *appropriate* (and only four of them intersect with the candidate space). This example highlights the importance of modelling productive difficulties for test types with open answer format.

For the closed cloze test, the candidate space is constant. The learners seem to consider even fewer options, on average only three of the five provided answers are actually selected. For closed formats, it is thus more relevant to model candidate ambiguity. In the following section, we analyze if the difficulty prediction can be performed for all test types despite the varying candidate space.

4 Difficulty prediction

Teachers are often not able to correctly anticipate the difficulties a learner might face. For the example in Section 2, one would probably expect high error rates for *vaccines* and *penicillin*, while the problems with *likely* and *that* might come as a surprise (see Figure 5). For optimal learner support, it is important to predict these difficulties.

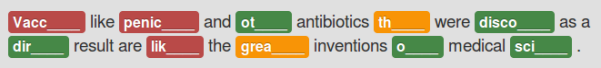

Figure 5: Visualization of gap difficulty. Easy gaps are marked green, intermediate gaps yellow and difficult gaps red.

4.1 Previous work

The earliest analyses of test difficulty operate on the level of the full text instead of individual gaps. Klein-Braley (1984) performs a linear regression analysis with only two difficulty indicators – average sentence length and type-token ratio – and obtained useful predictions of the mean test difficulty for her target group. Eckes (2011) also focuses on the mean test difficulty and aims at calibrating C-tests using a Rasch model to build a test pool.

Kamimoto (1993) performs classical item analysis on the gap level and creates a tailored C-test that only contains selected gaps which better discriminate between students. However, the gap selection is based on previous test results instead of gap features and cannot be applied on new tests.

In previous work (Beinborn et al., 2014a), we reported the first results for automatic difficulty prediction on the gap level. We introduced a model for the difficulty prediction of English C-test gaps that combines aspects of text and word difficulty with properties of the candidate space and gap dependencies. As the current work builds on this model, we summarize the feature space below.

Text difficulty For all test types, the difficulty of the test text determines the available context for the participant. A more challenging text increases the difficulty of all gaps as the participant's orientation in the text becomes more complicated (compare Brown (1989)). The difficulty of the underlying text can be determined by readability features. Our approach combines traditional features as the average sentence and word length with more advanced features from all linguistic levels (e.g. lexical, syntactic, semantic, discourse) including features specific to readability for language learning as for example

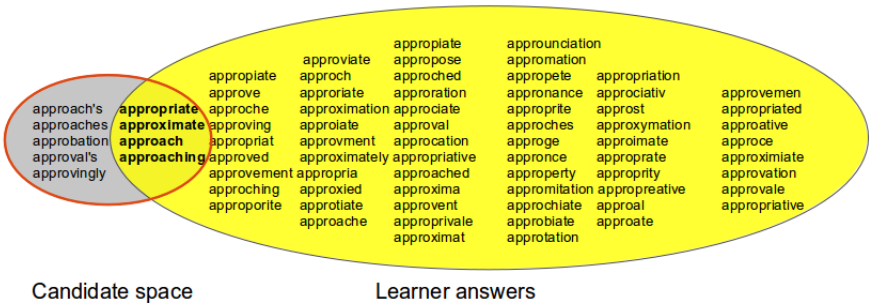

Figure 4: The candidates for the gap *appro___*: The computational candidate space is on the left, the answers provided by the students are on the right.

the "cognateness of words" (Beinborn et al., 2014b).

Word difficulty On a more local level, the difficulty of the particular solution word determines whether the participant is even capable of knowing the solution. Previous work by Brown (1989) and Sigott (1995) shows that the word class and the word frequency correlate with the gap difficulty. In addition, Klein-Braley (1996) analyzed error patterns in C-tests that are related to production problems (correct word stem in wrong form) and early closure (the participant selects a solution that works locally, but not in the larger context). In previous work, we operationalized and extended these findings by implementing many word difficulty features targeted at morphology, word frequency, cognateness and spelling (Beinborn et al., 2014a).

Test parameters In our previous work, we focused on English C-Tests, so the test parameters were fixed to a deletion rate of two, the C-test gap type and an open answer format. In order to evaluate the candidate space, we calculated the number of candidates and ranked them based on their frequency in the context. Due to the high deletion rate, the context is limited to the direct left and right neighbor. In addition, we evaluated the position of the gap and the solution frequency of the previous gap to assess dependencies between items.

4.2 Our generalized approach

Our approach for automatic difficulty prediction builds on the model introduced earlier for English C-tests as summarized above (Beinborn et al., 2014a). We reduce the feature set (from 87 to 70) and only include those features that can be adapted to all our target languages and test types. This comprises features estimating the difficulty of the solution and its direct context (cognateness, frequency, word class, inflection, compound structure, spelling difficulty, etc.)[4], the readability of the text (type-token-ratio, number of clauses, average word and sentence length, etc.) and the test parameters (number of candidates, position of gap, etc). We also adapt the pipelines to include proper German and French pre-processing using DkPro (de Castilho and Gurevych, 2014) and adapt the candidate calculation to the different test types.

In order to assure comparability to previous work, we also use support vector machines for regression in Weka (Hall et al., 2009) through the DKPro TC framework (Daxenberger et al., 2014).

4.3 Data

Table 2 provides an overview of the test datasets used in this paper. It consists of four open formats and one closed format with multiple choice options. The number of participants is averaged over the texts because each participant worked with 5 texts in the open formats.[5] The error rates should not be compared across test types because the participants had different levels of language proficiency. The high standard deviations indicate that each test contains gaps that are rather easy and others that are extremely difficult. In Beinborn et al. (2014a) we have shown that the error rate for a single item remains

[4]For the English C-test, we consider the cognateness to German because it is the L1 of the majority of the participants. For the French and German tests, we do not have information about the L1. We therefore consider the cognateness with English because we assume that most participants have at least rudimentary knowledge of English.

[5]The total number of participants is thus much higher, but this is not relevant for the error rate calculation

Format	Test type	Texts	Gaps	Particip.	Avg. error rate
Open	C-test en	39	775	210	.35±.25
	C-test fr	40	799	24	.52±.28
	C-test de	82	1,640	251	.55±.26
	Prefix de	14	348	225	.36±.23
Closed	Cloze en	100	100	22	.27±.22

Table 2: Overview of test data

quite stable for varying sample sizes.

C-test We use the same English C-test data as in our previous work (Beinborn et al., 2014a) and additionally obtained French tests. In both cases, the tests served as a placement test at the language centre of the TU Darmstadt in order to assign students to language levels. The participants had heterogeneous backgrounds regarding their language proficiency and mother tongue, but the majority was German. Furthermore, we received German C-tests from the TestDaf institute that have been administered to foreign students who apply for studying in Germany. It is a subset of the data described in Eckes (2011).

Prefix deletion For the prefix deletion test, we received German tests from the University of Duisburg-Essen that test the proficiency of prospective teachers.[6] The participants are a mix of native German speakers and students with migratory background (26%). Their language proficiency is much higher than that of the participants in the other tests.

Cloze tests For cloze tests, we could not find any test data with error rates. We thus conducted a study to collect error rates ourselves using the Microsoft sentence completion dataset.[7] For this dataset, Zweig and Burges (2012) transformed 1400 sentences from 5 Sherlock Holmes novels (written by Arthur Conan Doyle) into cloze tests. In each selected sentence, they replace a low-frequency content word with a gap and provide the solution along with 4 distractors (so-called closed cloze). The distractors were generated automatically based on n-gram frequencies and then handpicked by human judges. It should be noted that all distractors form grammatically correct sentences and that the n-gram probabilities for the answer options are comparable.

[6] http://zlb.uni-due.de/sprachkompetenz

[7] http://research.microsoft.com/en-us/projects/scc/, 15.12.2014

Dataset	LOO Gaps	LOO Texts
C-test en	.55	.47
C-test fr	.70	.67
C-test de	.63	.61
Prefix de	.54	.27
Cloze en	.20	.20

Table 3: Pearson correlation for difficulty prediction results in an leave-one-out cross-validation setting on the gap and on the text level

We tested a subset of the cloze questions with an eloquent native speaker of English and he answered 100% correctly. In order to determine the difficulty for language learners, we set up 10 web surveys with 10 questions each (as in Figure 1) and asked advanced learners of English to answer them.

4.4 Prediction Results

Table 3 shows the correlation between the measured human error rates and the predictions of our generalized prediction approach. It should be noted that we used the same features for each dataset. In practical applications, it would of course be possible to tune the feature selection for each task separately. For research purposes, however, we are interested in creating uniform conditions to allow a more meaningful comparison.

In our previous approach, we performed leave-one-out testing on all gaps to account for the small amount of training data. As each text of the open format test types contains 20 gaps, leave-one-out testing on all gaps increases the risk of over-fitting the model to specific text properties. For a more realistic prediction setting, we additionally perform leave-one-out testing on the texts, i.e. we always test on 20 gaps from one coherent text. We will focus on the results reported for this scenario, although they are slightly worse. The baseline, that always predicts the mean error rate, yields a correlation of 0 for all test types.

Languages The results show that the difficulty prediction can be successfully adapted to other languages. The correlation for the English C-tests is a bit lower than in previous work (0.60) because we reduced the set of features as described above. This allowed us to obtain results for German and French that are even better than the ones previously reported for English.

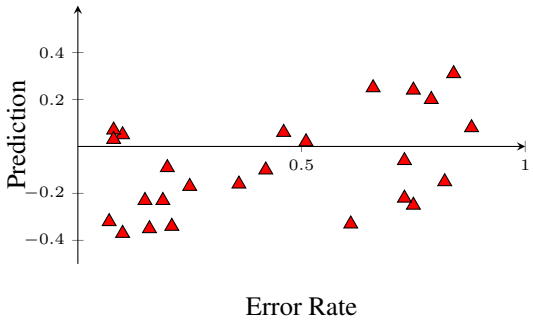

Figure 6: Biased prediction for the outlier text in the prefix deletion dataset

Test Types The results for the test types show that the prediction framework struggles with the prefix deletion and the cloze tests. One obvious reason could be the size of the training data which is significantly smaller for these tasks.

We first have a closer look at the prefix deletion test to explain the strong decline for leave-one-out cross-validation on texts. We find that the most significant prediction errors can be found for one particular text. This text exhibits a very high readability (e.g. low type-token and pronoun ratio, few adjectives and adverbs), but contains many difficult gaps. This combination has not been observed in the training data which explains that the difficulty of all gaps is strongly underestimated (resulting in negative values for the predicted error rates).

Figure 6 shows that the differences between gaps are actually predicted quite well, one could simply add a constant factor (of about 0.4) to receive an acceptable prediction. For the purpose of the error analysis, we remove that particular text from the evaluation and re-calculate the results. This yields a more reasonable Pearson correlation of 0.43 and shows that the difference between LOO on gaps and on text is due to over-fitting to text properties of the training data. This effect would surely decrease with more training data as can be seen for the bigger French and German datasets.

For the cloze test on the other hand, something more essential is going wrong. In Section 3, we have seen that the main difference for this test type is the closed candidate space. The features modelling production problems are thus not relevant here. While the number of the candidates is fixed, the set is still very variable because the distractors can be freely selected from the whole vocabulary. The better the

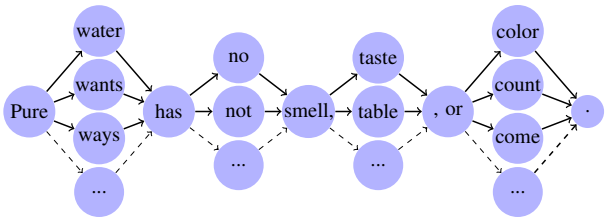

Figure 7: The search space for the sentence *Pure wa___ has n___ smell, ta___, or co___*. In this graph, the solution is always the topmost candidate, the candidate space is simplified.

distractors fit the gap, the more difficult it gets for the learner to select the solution, as in the following example:

When his body had been carried from the cellar we found ourselves still confronted with a problem which was almost as ____ as that with which we had started.

[tall, loud, invisible, quick, formidable]

Only very few learners managed to identify the solution *formidable* in this case, while the example in Figure 1 was quite easy for them. For difficulty prediction, it is therefore important to estimate the ambiguity of the answer options. In the remainder of the paper, we examine whether strategies that have been successfully applied for automatic solving of language tests can also provide insights into human difficulties with candidate ambiguity.

5 Candidate evaluation strategies

The main challenge for solving a reduced redundancy test consists in identifying the most suitable candidate in the candidate space. The context fitness of a candidate can be evaluated based on language model probabilities and on semantic relatedness between the candidate and the context.

LM-based approach A probabilistic language model (LM) calculates the probability of a phrase based on the frequencies of lower order n-grams extracted from training data (Stolcke, 1994). This can be used to predict the fitness of a word for the sentential context. Bickel et al. (2005), for example, evaluate the use of probabilistic language models to support auto-completion of sentences in writing editors. In the completion scenario, only the left context is available, while the learner can also consider the right context in language tests. Zweig et al. (2012)

thus model the problem of solving cloze tests by applying methods from lexical substitution to evaluate and rank the candidates. The part to be substituted is a gap and the set of "substitution candidates" is already provided by the answer options.

Unfortunately, we cannot rely on static sentences for the open test formats as the context needs to be determined by solving the surrounding gaps. For each gap, we take all candidates into account and generate all possible sentences resulting from the combinations with the candidates of subsequent gaps. This can lead to strong dependencies between items, i.e. solving a subsequent item is facilitated, if the previous one has been solved correctly. As a consequence, we need to evaluate a combinatorial search space that grows exponentially with the number of gaps in the sentence (see Figure 7). We thus use a pruning step after each gap that scores the generated sub-sentences using a language model and only keeps the n best. For the closed cloze test, the number of generated sentences is of course limited to the number of candidates (5) because each sentence contains only one gap.

We use 5-gram language models that are trained on monolingual news corpora using berkeleylm with Kneser-Ney smoothing.[8] Zweig et al. (2012) trained their models explicitly on training data only from Sherlock Holmes novels. In order to better simulate learner knowledge, we use rather small and controlled training data from the Leipzig collection (Quasthoff et al., 2006) consisting of one million sentences for each language.

For solving the test, we then select the generated sentence with the highest log-probability in the language model and count how many gaps are solved correctly. If several sentences obtain the same probability, we pick one at random. We run this strategy ten times and average the results. For comparison, we implement a baseline that always selects the most frequent candidate without considering the context.

Semantic relatedness approach Language models cannot capture relations between distant words in the sentence. To account for this constraint, Zweig et al. (2012) include information from latent semantic analysis (Deerwester et al., 1990). For this method, every word is represented by a vector of related words that is calculated on the basis of training data. The semantic relatedness between two words can then be expressed by the cosine similarity of the two vectors. Similar to Zweig et al. (2012), we sum over the cosine similarity between the candidate and every content word in the sentence to calculate the candidate fitness. While they calculate relatedness based on a latent semantic analysis index of the domain-specific Holmes corpus, we use explicit semantic analysis (Gabrilovich and Markovitch, 2007) calculated on Wikipedia to better model the learner's general domain knowledge.[9] The semantic approach cannot be applied on open formats because semantic relatedness is not informative for function words and inflections.

[8] http://code.google.com/p/berkeleylm, 15.12.2014

	Human	Baseline	LM-Based	Semantic
C-test en	.68	.11	.76	-
C-test fr	.48	.10	.79	-
C-test de	.45	.09	.76	-
Prefix de	.64	.09	.73	-
Cloze en	.70	.21	.26	.32

Table 4: Solving accuracy for the different candidate evaluation strategies

Results The accuracy of the automatic solving strategies and the average human performance in Table 4 shows that the LM-based solving strategy strongly outperforms the baseline and can also beat the average human solver for the open test formats.[10] Even the large candidate space of the prefix deletion test can be disambiguated quite well. For the cloze tests, the candidate ambiguity seems to be more challenging. The LM-based candidate evaluation only performs slightly better than the baseline due to the fact that the distractor generation approach assured comparable context frequency of all candidates. The semantic relatedness approach works slightly better, but also fails to select the correct candidate in most cases.

Not surprisingly, our results for the cloze tests are worse than those obtained with domain-specific corpora in previous work. However, we are not interested in developing a perfect solving method, but aim at modelling the difficulty for the learner. A

[9] Index retrieved from https://public.ukp.informatik.tu-darmstadt.de/baer/wp_eng_lem_nc_c.zip, 30.03.2015

[10] The human results should not be compared across test types as the participant groups had different backgrounds and different language proficiency.

question is less likely to be solved if the context fitness of a distractor is rated higher than that of the solution. The failures of the automatic solving might hence be indicative for the difficulty prediction for cloze tests.

6 Improved difficulty prediction

The solving approaches described above provide a ranking of the candidates that can be instrumental for difficulty prediction. We develop two new features that evaluate the context fitness of the candidates based on the measures described above and return the rank of the solution. We assume that a gap is more difficult if the solution is not the top-ranked candidate.

We have seen that many of the difficulty features that have been developed for the C-test are not applicable for the cloze data. The C-test difficulty has been modelled by estimating the size of the candidate space (which is constant in this case), production difficulties (which are not relevant in closed formats), and a frequency-based ranking of the candidates (which has been controlled by the test designers). The remaining features measure the readability of the text, the frequency of the direct context, and the word class of the gap and provide important information about the general difficulty of the gap independent of the answer options. We analyze if the ranking features can then capture the important aspect of candidate ambiguity to improve difficulty prediction for cloze tests.

Results The results in Table 5 show that reducing the feature set to those that are actually relevant for closed formats already has a small effect, but it is not significant. Adding the ranking features then leads to a strong improvement in difficulty prediction. The best result is obtained with the semantic relatedness ranking.

We explained above that the LM-based approach is not suitable for solving this cloze dataset because the answer options have been controlled with respect to frequency. However, the participants are not aware of this constraint, and frequency effects actually do play a role in learner processing. This explains that LM-based ranking can also be beneficial for difficulty prediction.

Our results show that modelling the context fitness of the candidates is essential for predicting the difficulty of closed cloze tests.[11]

	# Features	Pearson's r
Standard features	70	.20
Reduced features	33	.24
Reduced + LM ranker	34	.38*
Reduced + Semantic ranker	34	**.42***
Reduced + LM + Semantic ranker	35	.39*

Table 5: Improved prediction results for cloze tests. Significant differences to the result with the standard features are indicated with * ($p<0.01$).

7 Conclusions

In this work, we have performed difficulty prediction for different types of reduced redundancy testing for several languages. To our knowledge, this is the first approach to predict the difficulty of prefix deletion tests, cloze tests and French and German C-tests. We obtained remarkably good results for French and German that were even better than the ones previously reported for English. In practical teaching scenarios, the feature selection could be further tuned to the respective test type and learner group.

In order to improve difficulty prediction for closed test formats, we developed two ranking strategies for candidate evaluation inspired by automatic solving methods. The approaches evaluate the fitness of a candidate in the sentential context based on language model probability and semantic relatedness. We have reached significant improvements of the difficulty prediction for closed cloze tests by including these ranking features. Especially the semantic approach seems to be a good model for human evaluation strategies.

For future work, we will extend our analysis to a bigger set of closed test formats and work towards better models of learner knowledge.

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.

[11] For the open test formats, the additional features had almost no effect.

References

Manish Agarwal and Prashanth Mannem. 2011. Automatic Gap-fill Question Generation from Text Books. pages 56–64. Association for Computational Linguistics.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014a. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–529.

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

Steffen Bickel, Peter Haider, and Tobias Scheffer. 2005. Predicting sentences using N-gram language models. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pages 193–200, Morristown, NJ, USA, October. Association for Computational Linguistics.

Aleid C Broerse and EJ Zwaan. 1966. The information value of initial letters in the identification of words. *Journal of Verbal Learning and Verbal Behavior*, 5(5):441–446.

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.

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, Baltimore, Maryland, June. 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. *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

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

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611.

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

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.

Tobias Horsmann and Torsten Zesch. 2014. Towards automatic scoring of cloze items by selecting low-ambiguity contexts. *NEALT Proceedings Series Vol. 22*, pages 33–42.

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.

Sachiko Kinoshita. 2000. The left-to-right nature of the masked onset priming effect in naming. *Psychonomic Bulletin & Review*, 7(1):133–141, March.

Christine Klein-Braley and Ulrich Raatz. 1982. Der C-Test: ein neuer Ansatz zur Messung allgemeiner Sprachbeherrschung. *AKS-Rundbrief*, 4:23 – 37.

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.

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.

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. pages 136–146.

Laura Perez-Beltrachini, Claire Gardent, and German Kruszewski. 2012. Generating Grammar Exercises. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 147–156.

Uwe Quasthoff, Matthias Richter, and Christian Biemann. 2006. Corpus portal for search in monolingual corpora. In *Proceedings of the fifth international conference on language resources and evaluation*, volume 17991802.

Günther Sigott and Johann Köberl. 1996. Deletion patterns and C-test difficulty across languages. In Rüdiger Grotjahn, editor, *Der C-Test. Theoretische Grundlagen und praktische Anwendungen 3*, pages 159–172. Brockmeyer, Bochum.

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

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)*. Association for Computational Linguistics.

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.

Andreas Stolcke. 1994. *Bayesian learning of probabilistic language models*. Ph.D. thesis, University of California, Berkeley.

Wilson L. Taylor. 1953. "Cloze procedure": a new tool for measuring readability. *Journalism Quarterly*, 30:415–433.

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.

Geoffrey Zweig and Chris JC Burges. 2012. A challenge set for advancing language modeling. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 29–36. Association for Computational Linguistics.

Geoffrey Zweig, John C Platt, Christopher Meek, Christopher JC Burges, Ainur Yessenalina, and Qiang Liu. 2012. Computational approaches to sentence completion. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 601–610. Association for Computational Linguistics.