Automatic Identification of Nutritious Contexts for Learning Vocabulary Words

Jack Mostow, Donna Gates, Ross Ellison, Rahul Goutam
Project LISTEN (www.cs.cmu.edu/~listen), School of Computer Science, Carnegie Mellon University
RI-NSH 4103, 5000 Forbes Avenue, Pittsburgh, PA 15213-3980, USA
011 (412) 268-1330
mostow@cmu.edu, dmg@alumni.cmu.edu, rpelliso@andrew.cmu.edu, rgoutam@cmu.edu

ABSTRACT

Vocabulary knowledge is crucial to literacy development and academic success. Previous research has shown learning the meaning of a word requires encountering it in diverse informative contexts. In this work, we try to identify "nutritious" contexts for a word – contexts that help students build a rich mental representation of the word's meaning. Using crowdsourced ratings of vocabulary contexts retrieved from the web, AVER learns models to score unseen contexts for unseen words. We specify the features used in the models, measure their individual informativeness, evaluate AVER's cross-validated accuracy in scoring contexts for unseen words, and compare its agreement with the human ratings against the humans' agreement with each other. The automated scores are not good enough to replace human ratings, but should reduce human effort by identifying contexts likely to be worth rating by hand, subject to a tradeoff between the number of contexts inspected by hand, and how many of them a human judge will consider nutritious.

Keywords

Vocabulary learning, crowdsourcing, automated scoring, regression models.

1. INTRODUCTION

Years of research on vocabulary learning have shown that vocabulary is a bottleneck to comprehension [1], demonstrated that vocabulary instruction benefits students' word learning and comprehension of text [2-5], and identified several principles of effective vocabulary instruction [6-12]. The principle relevant to this paper is that vocabulary learning requires exposure to diverse informative example contexts in order to develop a rich mental representation of word meanings and their relations to other words.

This paper describes AVER ("Automatic Vocabulary Example Rater"), an attempt to automatically identify "nutritious" contexts – example uses of a word that should help in learning its meaning. This work is part of a larger project that supplied our training and test data in the form of target vocabulary words, example contexts in which they occur, and human ratings of their nutritiousness. The contexts were retrieved from the web by DictionarySquared.com, an online high school vocabulary tutor that searches the web for a given target word in order to find candidate contexts that contain it. DictionarySquared aims to pick contexts a few dozen words long, preferring to start and end at boundaries between sentences, paragraphs, or HTML blocks.

This paper describes how AVER trains and evaluate models to predict the nutritiousness of such contexts, based on human ratings crowdsourced using Amazon Mechanical Turk.

Ideally AVER would identify a set of examples that maximizes the amount of actual student learning from a given number of contexts, taking into account the diversity of multiple contexts for the same word, and possibly even their relation to example contexts for other target vocabulary words to learn. However, this paper focuses on the initial problem of predicting the suitability of individual contexts, using crowdsourced human estimates instead of students' subjective ratings of contexts, or objective measures of their actual learning gains.

1.1 Relation to Prior Work

Some previous work has addressed the problem of finding suitable example contexts to support vocabulary learning, but differed in one or more respects from the work reported here. REAP [13] selected examples from an already-vetted corpus, based on specified selection criteria such as student interests. VEGEMATIC [14] constructed 9-word contexts centered on a given target vocabulary word by concatenating overlapping 5-grams from the Google *n-gram* corpus, based on heuristic constraints and preferences; only some of them were good enough to use, but hand-vetting them was faster than composing good examples by hand. Follow-on work [15] extended VEGEMATIC to generate contexts for a particular sense of a target word. AVER also seeks to identify example contexts suitable for vocabulary learning, but addresses a different goal than both these projects: instead of applying explicit hand-crafted heuristics, AVER learns to predict crowdsourced ratings by human judges.

The rest of the paper is organized as follows. First we describe our data set. Then we describe the features we used, tried but dropped, or identified but didn't implement. Next we describe and evaluate how AVER rates contexts. Finally we conclude.

2. DATA SET

The data for this work consists of a vocabulary word and a context that contains at least one instance of the vocabulary word and that illustrates usage of the vocabulary word. The overall data set includes 75,844 contexts for 1,000 vocabulary words, comprising 100 words from each of 10 difficulty bands based on their Standardized Frequency Index [16], a measure of log frequency in a text corpus, adjusted by dispersion across multiple domains.

Dr. Margaret G. McKeown, an international expert on vocabulary learning and instruction, rated 93 contexts based on three criteria – the typicality of the usage of the vocabulary word in the context, the degree to which the context constrains the meaning of the vocabulary word, and the comprehensibility of the context for students. Thus the expert provided three ratings of each context, one on each criterion, ranging from 1 (very poor) to 5 (very good). These data helped in developing a rating scale. However, it would have been infeasible to obtain expert ratings of enough contexts to train good models.

Therefore, using Amazon Mechanical Turk, 13,270 contexts were each rated by 10 amateur raters who passed a brief test of their performance on this task: "Based on context, rate how helpful the text is for helping a high school student understand the meaning of the target word. A helpful context is one that reinforces a word's meaning and is understandable to high school students." Contexts ranged in length from 18 to 137 words, with median 63.

Raters differed in how many contexts they rated, ranging from several to hundreds. They rated contexts on a 5-point scale:

- 4 = Very Helpful: After reading the context, a student will have a very good idea of what this word means.
- 3 = Somewhat Helpful
- 2 = Neutral: The context neither helps nor hinders a student's understanding of the word's meaning.
- 1 = Bad: The context is misleading or too difficult.
- 0 = Otherwise inappropriate for high school students.

We used the mean of their 10 ratings to label our training and testing data. Inter-rater standard deviation averaged 0.81, so standard error averaged 0.27. We labeled the 4107 contexts with mean rating at or above 3 as "good," and the 9150 contexts with mean rating below 3 as "bad."

3. FEATURES USED

The remaining 62,574 contexts were not rated by humans. To rate their nutritiousness automatically, AVER uses the human-labeled data to train and test regression models to predict the ratings of unseen contexts for unseen words, or to predict the probability that a context is "good," i.e., its rating is greater than or equal to 3.

To train these models, we extract features of the vocabulary word and context we consider likely to be informative in predicting its human rating. We normalize every feature as a z-score by subtracting the mean value for that feature and dividing by its standard deviation. By translating all feature values onto a common scale, normalization makes their regression coefficients comparable. Normalization does not affect a feature's correlation with Turker ratings or other features because correlation is invariant under constant addition or multiplication. We assign a z-score of zero to features with undefined values, so that they have no impact on model output.

To describe various types of features, illustrate their values, explain their meaning, and discuss the intuition underlying them, we will use the following example context for the vocabulary word ***alleviate***, with mean Turker rating = 3.7, i.e. quite good:

> ***It is ironic that students are pressured to do well in school in order to continue participating in extracurricular activities, yet these after school activities are just what they need to relieve stress. Sports clubs and even being involved in student government can help alleviate stress. They allow us to get away from school pressure and enjoy ourselves.***

3.1 Comprehensibility

Our goal is to help students learn the typical usage of a vocabulary word by providing them with example contexts. If the example contexts are too difficult to understand, they will not be very helpful to students. Thus indicators of comprehensibility are useful features in predicting the rating of a context.

Rarer words are typically harder. The log frequency of ***alleviate***, i.e., the log of its unigram count (1,596,620) divided by the total number of tokens (1,024,908,267,229) in the Google *n-gram*s corpus, is –13.4 (z-score = –0.090), placing it in the third most common of 10 word bands (z-score = 0.150). This feature of the target word is the same for all its contexts, but helps control for target word frequency in general models to predict context ratings.

The more and longer the words in a context, the harder it is to understand. The example context has 58 words (z-score = –0.235, which on average are 5.1 letters long (z-score = 0.358), not counting spaces or punctuation.

Flesch-Kincaid scores for reading ease and grade level are widely used to assess readability, and we compute them for contexts:

Reading ease =

$$206.835 - 1.015 * \frac{total\ words}{total\ sentences} - 84.6 * \frac{total\ syllables}{total\ words}$$

Grade level =

$$0.39 * \frac{total\ words}{total\ sentences} + 11.8 * \frac{total\ syllables}{total\ words} - 15.59$$

A higher reading ease score characterizes text as easier to read and understand. The reading ease score ranges from 0 to 100. The reading ease score for our example context is 47.18, indicating that it is moderately difficult (z-score = –0.015). Flesch-Kincaid scores depend on how syllables, words, and sentences are counted, and hence differ from one implementation to another, but not by much. Thus Microsoft Word reports a reading ease of 48.6 for this paragraph.

A higher grade level score indicates a context that is more difficult to read and understand. The grade level roughly translates to the number of years of education required to understand the context. The grade level score for our example context is 11.48 (z-score = –0.217), compared to 11.2 in Microsoft Word.

Mean human ratings correlated 0.009 with log of target frequency, 0.023 with word band, –0.082 with context length, –0.039 with average word length, 0.043 with reading ease, and –0.030 with grade level.

3.2 Local Predictability

AVER extracts local predictability features from a 9-word context centered on the target word (e.g. ***student government can help alleviate stress . They allow***). They estimate the probability of the target word given a local context containing the target word. Five of these local contexts are 5 words long, four are 4 words long, three are 3 words long, two are 2 words long, and the target itself can be considered a 1-word context, so there are 15 probabilities. The submitted version of this paper used all 15 of these probabilities as features.

To estimate these probabilities, AVER uses the Google *n-gram*s tables [16] based on over a trillion words from the web. These tables specify the frequency of every word unigram, bigram, trigram, 4-gram, and 5-gram with at least 40 occurrences. Thus AVER can use them to estimate such conditional probabilities up to a context length of 5 words. For example, it would estimate the conditional probability of ***alleviate*** given the 5-word local context ***government can help ____ stress*** as the frequency of the 5-gram ***government can help alleviate stress*** divided by the summed counts of all 5-grams of the form ***government can help * stress.***

AVER log-transforms the probability estimates to reduce their enormous dynamic range, and normalizes the log probabilities as z-scores, which it uses as features to measure local predictability.

If the numerator is zero, AVER smoothes it to 1. The numerator is zero for 88% to 93% of the 5-word contexts, varying by the position of the target word. E.g., ***help alleviate stress . They*** is not in the 5-gram table. The numerator is zero for 68% to 78% of the 4-word contexts, 33% to 44% of the 3-word contexts, and 8% to 9% of the 2-word contexts.

What if the denominator is zero (e.g. no 5-grams of the form ***government can help * stress*** are listed in the 5-gram table)? The denominator is zero for 82% to 86% of our 5-word contexts that contain the target word; the percentage varies by its position in the context. Likewise, the denominator is zero for 47% to 57% of the 4-word contexts, and 33% to 44% of the 3-word contexts.

In the submitted version of this paper, we translated the resulting undefined probability into a z-score of zero, so that it would not neither increase nor decrease the output of our predictive models. However, the effect was that some features, especially for 5-grams, were mostly zero in the training data. Could we do better?

Inspired by a reviewer comment, we implemented a new version, called AVER.b (b for "backoff") based on an idea from statistical language modeling: in the absence of data about a particular *n-gram*, back off to successively shorter *n-gram*s. For instance, if the denominator is zero because no 5-grams of the form ***government can help * stress*** are in the 5-gram table, AVER.b looks for 4-grams of the form ***government can help ** or ***can help * stress.*** If AVER.b finds both, it backs off to whichever yields a higher probability for the target word, on the assumption that it is more informative. If it finds neither, it backs off to trigrams, then bigrams, then finally the unigram ***alleviate***.

For our example, contexts of the form ***can help * stress .*** are the only ones listed in the 5-gram table. The 4 listed contexts contain ***alleviate*** (109) ***reduce*** (455), ***relieve*** (329), and ***with*** (49). The numerator 109 and denominator 942 yield log probability –2.16.

For the other 4 positions, AVER.b backs off to 4-grams. Its 4-gram table yields non-zero denominators for 4-word contexts of the form ***help * stress .*** (4829), ***can help * stress*** (6484), and ***government can help ** (6765). It yields non-zero numerators for ***help alleviate stress .*** (330) and ***can help alleviate stress*** (325) but zero for ***government can help alleviate***, which it smoothes to 1, yielding respective log probabilities of –2.68, –2.99, and –8.82.

AVER.b finds no 4-grams of the form ***** stress . They***, so it backs off to 3-grams, using the count of ***alleviate stress .*** (2120) as numerator and the number of 3-grams of the form ***** stress .*** (1599767) as denominator, yielding log probability -6.63.

To speed up such computations, we had years earlier indexed each table by various sequences of *n-gram* positions designed to quickly retrieve all rows matching the values specified for any subset of positions. Table 1 lists these indexes, which took weeks of computer time to build because the tables have so many rows.

Table 1: Indexes constructed for Google *n-gram*s tables

Table:	# rows:	Indexed by:
unigram	13,588,391	1, frequency
bigram	314,843,401	12, 21
trigram	977,069,902	123, 312, 23
4-gram	1,313,818,354	1234, 234, 314, 412, 24, 34
5-gram	1,176,470,663	12345, 5432, 3145, 2541, 1523, 432

For instance, to look up the count of the 5-gram ***government can help alleviate stress*** efficiently, both versions of AVER uses the index 12345. This count is the numerator for estimating the probability of ***alleviate*** at word 4 given a 5-word context. To find all 5-grams of the form ***government can help * stress***, AVER uses the index 1523. If it finds any, it sums their frequencies as the denominator. If not, AVER.b backs off as described above. It then uses the index 1234 to look up the 4-grams ***government can help alleviate*** and ***can help alleviate stress*** and 4-grams of the form ***government can help ****, and the index 412 to find 4-grams of the form ***can help * stress***.

This method if necessary estimates the conditional probability of ***alleviate*** given the local bigram context ***help*** ____ as the bigram frequency of ***help alleviate*** divided by the summed frequency of all bigrams of the form ***help ____.*** However, there are 28,578 bigrams of this form, and it takes non-trivial time to retrieve them in order to compute their summed frequency of 270,480,813. Instead, both versions of AVER would approximate this sum as the unigram frequency of ***help***, namely 271,840,666, which it can retrieve quickly from a single row of the Google unigram table. This over-estimate counts all bigrams of the form ***help ____*** that occurred fewer than 40 times in the Google *n-gram*s corpus and hence do not appear in the Google bigrams table. This approximation is possible only if the blank falls at the start or end of the *n-gram*. Thus it can approximate the number of trigrams of the form ***can help ** or ***** stress .***, but not ***help * stress***. The approximation was not necessary for 4- or 5-grams because they typically have many fewer rows in the *n-gram* table.

A target word can occur at *n* different positions in a word window of size *n*, with a separate probability for each window size and position within the window, represented as a log probability. Consequently, original AVER's local predictability features consist of 1 + 2 + 3 + 4 + 5 = 15 different log probabilities. For our example context, their respective z-scores are –0.090; –0.120, 0.740; 0.431, 1.340, –6.775; 0, 0.972, 0.909, –0.351; and 0, 0, 0.603, 0, 0. The z-scores of zero reflect the sparsity of *n-gram*s as *n* increases.

The relative weights of these 15 z-scores reflect the overall local predictability of the target word ***alleviate*** in the local context ***student government can help alleviate stress . They allow***. AVER sets these weights empirically as part of optimizing the weights for all our features, not just these 15. Correlations of the 15 features with human ratings range from 0.138 for logP(target w1 | ___ w1) to –0.009 for logP(target w1 w2 w3 w4 | ___ w1 w2 w3 w4). That is, the word ***stress*** makes it likelier that ***alleviate*** precedes it, but ***stress . They allow*** makes ***alleviate*** slightly less likely.

In contrast, AVER.b uses just 5 local predictability features, one for each position in a 5-word context. In our example, their respective values are 0.071, 1.006, 1.157, 0.944, and –0.457. The third value is largest, i.e. ***can help __ stress .*** is the 5-word context that most strongly predicts ***alleviate***. These 5 features correlate with mean Turker ratings at 0.055, 0.038, 0.065, 0.042, and 0.062.

To estimate the probability of the target word at word *i* given a 5-word window, AVER.b uses *n-gram*s whose length n_i varies by the amount of backoff. To reflect the relative specificity of the evidence for each estimated probability, we tried weighting it by

$$n_i / \sum_{i=1}^{5} n_i$$

but it made model fit slightly worse, so we decided not to weight by *n-gram* length. Perhaps weighting it differently would help.

3.3 Topicality

Topicality features measure relatedness of the target vocabulary word to other content words in the context. The intuition behind using such features is that a context containing a typical usage of the target vocabulary word is likely to contain other content words that co-occur frequently with the target vocabulary word or are distributionally similar to it, i.e. tend to co-occur with the same words that the target word co-occurs with. The DISCO tool [17] at www.linguatools.de measures the co-occurrence of two words within 3 words of each other ("S1") and their distributional similarity ("S2") in a specified corpus, such as the British National Corpus, which contains 119 million tokens and 122,000 unique content words in "samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of British English, both spoken and written, from the late twentieth century" [18]. AVER uses DISCO to compute co-occurrence and distributional similarity between the target vocabulary word and each other content word in the context.

To score the overall topicality of a context for the target word, we must aggregate the relatedness scores for the individual context words. Typically only a few of the context words are strongly related to the target word. Consequently, the overall average relatedness of the context dilutes their influence. Instead, AVER averages relatedness over just the most related ***k*** words of the context. In informal tests of different values of ***k***, the average of the top 5 relatedness scores did best at predicting human ratings.

Thus AVER computes two topicality scores for a context. The co-occurrence z-score for our example context is 5.063. Context words that tend to co-occur with the target vocabulary word ***'alleviate'*** include ***'pressure'*** and ***'stress'.*** The distributional similarity z-score for our example context is 1.497. The context word with the highest distributional similarity to ***'alleviate'*** is ***'relieve'.*** DISCO's S1 and S2 scores based on BNC correlated with mean human context ratings at 0.060 and 0.025, respectively.

4. FEATURES TRIED BUT ABANDONED

We now discuss several features that we experimented with but do not use in AVER, either because they hurt predictive accuracy in informal small experiments, or because they were too complex to compute efficiently.

4.1 Topicality Based on Google *N-gram*s

As explained above, AVER computes context typicality using DISCO co-occurrence and similarity scores based on the 112 million word British National Corpus. These scores suffer from data sparsity in the case of less-frequent words. In contrast, the Google *n-gram*s corpus is based on over 10,000 times as much text, namely a trillion words of Web text. Not only is this corpus four orders of magnitude larger than BNC, it is also more relevant to the example contexts because they too consist of Web text.

Although the Google *n-gram*s corpus is already in the form of *n-gram*s rather than the text they are based on, its size makes it computationally expensive to compute similarity scores from it, so in previous work we had precomputed and indexed a table of the number of *n-gram*s containing a given pair of words at a distance of 1, 2, 3, or 4 words, and those *n-gram*s' summed frequency. However, this table has 921,643,327 rows. Despite efficient indexing, a target word's co-occurrences take considerable time to look up – over 30 seconds for ***alleviate***. To compute distributional similarity with reasonable speed, we therefore estimated it from the first few hundred rows. Unfortunately, the resulting feature harmed rather than helped model accuracy. To compute more predictive estimates of co-occurrence and distributional similarity based on Google *n-gram*s, it might help to sample them more judiciously, and to adjust better for differences among target words to make estimates comparable.

4.2 Language Model Probability

To quantify the likelihood of a given context occurring in English, we used a language model trained on English text using the NLTK language model package at www.nltk.org. The motivation for this feature was to penalize contexts that contain ill-formed or incomplete sentences. We dropped this feature because it did not improve predictive accuracy, but maybe other variants of it might.

4.3 Weighted Human Ratings

Apart from different features that we tried out but did not include in the final model, we also investigated methods to improve the accuracy of the labels computed by averaging 10 raters' ratings of each context. These methods weighted the average based on each rater's degree of agreement with expert ratings of other contexts. The more closely the rater agreed with the expert on the contexts they both rated, the more accurately we expected the rater to rate contexts that the expert did not rate.

However, most raters did not overlap with the expert in terms of which contexts they rated. We therefore extended the method transitively to rate such raters based on their degree of agreement with raters who had no***n***-zero overlap with the expert, and on how closely those raters agreed with the expert on the contexts they both rated.

We also used the overlapping contexts to train a model to predict a rater's ***expected*** degree of agreement with the expert, based on features of the rater such as the total number of contexts he or she had rated. We hoped to use this model to predict agreement with the expert even for raters with zero overlap. However, the expert rated only 93 contexts, so very few raters overlapped with the expert. Even they overlapped too little to accurately estimate the rater's agreement with the expert. We therefore abandoned the approach of rating raters by their actual or expected agreement with the expert, and using it to weight the individual ratings averaged to rate a given context. Rating raters might be effective given a larger sample of expert ratings, and greater overlap of the raters with the expert.

5. FEATURES FOUND BUT NOT USED

Based on expert linguistic analysis of over 200 contexts whose human and automated ratings differed drastically, we identified

some syntactic and semantic features not exploited by the current models, and likely to improve them.

5.1 Syntactic Features

Additional syntactic features of a context could be computed by parsing it with the Stanford parser, and extracting them from the parse tree with Tsurgeon and Tregex, using the tools at nlp.stanford.edu/software/corenlp.shtml [19]. commondatastorage.googleapis.com/books/syntactic-ngrams/index.html [20] is a corpus of syntactic *n-gram*s that provides counts of dependency tree fragments, which could be used to rate the plausibility of the parse and to infer likely dependency relations among context words. If for some reason part-of-speech tagging the context is feasible but parsing it is not, its dependency relations could be inferred from its part-of-speech *n-gram*s [21].

Informative syntactic features include the direct object of a target verb, e.g. ***abdicate*** in ***Edward abdicated the throne***, and the objects of prepositions following a target word, e.g. ***keen*** in ***They are very keen on education***. Another syntactic feature comes from coordinate constructions, e.g., ***it is characterized by inconsistency and vagary***. The coordinated conjuncts are likely to be semantically similar or even synonymous.

It might also be useful to incorporate syntactic information into the current *n-gram* features. In particular, disaggregating *n-gram* features by the target word's part of speech in the context would exploit systematic statistical differences between parts of speech. For instance, if the target word is a verb, its subject is likely to precede it, and shed semantic light on what sorts of agents can perform the verb. Conversely, if the target word is an adjective, the noun phrase after it illustrates what the adjective can modify.

5.2 Semantic Features

Our analysis of misrated contexts found that spuriously low similarity ratings are often caused by lack of co-occurrences due to sparse data for less-frequent words. This deficiency might be addressed by augmenting BNC data with definitions, Wordnet gloss examples, and Google *n-gram*s, provided the computational issues discussed earlier are satisfactorily addressed. For example, if we use Google *n-gram* features only where BNC data is too sparse, they might not pose such computational bottlenecks. Likewise, we could complement DISCO metrics of semantic similarity with features based on WordNet links from a target word to any of its synonyms, antonyms, hypernyms, and hyponyms that occur in the context.

6. AUTOMATED RATING OF CONTEXTS

AVER and AVER.b use the features described above in two types of models to rate contexts automatically for a given target word. The linear regression model predicts the mean human rating of a context. The logistic regression model is a binary classifier: it predicts whether a context is "good" (rated 3 or above) or "bad" (below 3).

We could run these models on all 75,844 contexts, but we can evaluate the models only on the 13,270 contexts rated by humans. To estimate the performance of both models on unseen data, we therefore use 5-fold cross-validation: We split the target words randomly into 5 equal subsets so as to partition the contexts into 5 subsets ("folds") with no overlap in target words between folds. For each fold we train both models on the other 4 folds, measure their performance on the held-out fold, and average over the held-out folds to estimate predictive accuracy on unseen target words – including the 62,574 unrated contexts, assuming they're similar.

To estimate performance fairly on unseen target words, it is essential to avoid overlap in target words between folds. Otherwise even if contexts do not overlap across folds, overlap in target words causes overfitting and inflates estimated performance on unseen data, especially if the training and test sets contain very similar contexts. Our initial results suffered from this problem before we eliminated overlap in target words across folds.

For the original AVER, the correlation between predicted and actual mean human ratings is 0.180 for the linear model and 0.178 for the logistic model. The Area Under Curve (AUC) for the original AVER is 0.600, significantly better than the 0.5 expected from a random baseline.

The linear model predicts mean human ratings, so it optimizes the correlation of predicted to actual ratings. The logistic model classifies contexts as good or bad, so it optimizes the number of misclassified contexts. Consequently correlation is higher for the linear model, whereas AUC is higher for the logistic model.

Unfortunately, AVER.b fared considerably worse. Its predictions correlated with actual ratings at only .093, with AUC only 0.563. Accordingly we focus on the results for the original AVER.

Table 2 shows the original AVER linear model's coefficients for each normalized feature. According to this analysis, the features in **boldface** are reliable at $p < .05$ (*), .005 (**), or .0005 (***).

Table 2: Coefficients of linear model for (original) AVER

Feature	Coefficient
WordBand	–.5691
Flesch-Kincaid Reading Ease	*** **.1220**
Flesch-Kincaid Grade	*** **.0627**
Average word length	*** **.0520**
Unigram logP(t)	* –1**.017**
Bigram logP(t w1 | __ w1)	*** **.0621**
Bigram logP(w1 t | w1 __)	** **.0188**
Trigram logP(t w1 w2 | __ w1 w2)	*** **–.0394**
Trigram logP(w1 t w2 | w1 __ w2)	.0070
Trigram logP(w1 w2 t | w1 w2 __)	*** **–.0053**
4gram logP(t w1 w2 w3 | __ w1 w2 w3)	.0088
4gram logP(w1 t w2 w3 | w1 __ w2 w3)	.0213
4gram logP(w1 w2 t w3 | w1 w2 __ w3)	–.0109
4gram logP(w1 w2 w3 t | w1 w2 w3 __)	*** **.0398**
5gram logP(t w1 w2 w3 w4 | __ w1 w2 w3 w4)	* **–.0297**
5gram logP(w1 t w2 w3 w4 | w1 __ w2 w3 w4)	–.0002
5gram logP(w1 w2 t w3 w4 | w1 w2 __ w3 w4)	.0193
5gram logP(w1w2w3 t w4 | w1 w2 w3 __ w4)	* **–.0283**
5gram logP(w1 w2 w3 w4 t | w1 w2 w3 w4 __)	.0017
Co-occurrence (DISCO S1)	*** **.0340**
Distributional Similarity (DISCO S2)	*** **.0674**
Intercept	*** **2.5079**

As Table 2 shows, unigram log probability of the target word was by far the most significant predictor of human ratings, and negative: contexts for rarer words get lower ratings, which may reflect that the less frequently the target word appears in the Google *n-gram*s corpus, the less likely it is to have good example contexts on the web. As expected, Reading Ease is a positive predictor: readable example contexts are likelier to help students. Surprisingly, the coefficients for word length and grade level are positive even though in isolation they correlate negatively with ratings. Perhaps they reflect positive effects exposed after other predictors account for the negative effects, or are simply artifacts of including correlated predictors in the model. Several *n-gram* based metrics of local predictability in the form of conditional probability of the target given the surrounding context are significant, but it is not clear why some are positive and others are negative. Fewer features based on longer *n-gram*s are significant, presumably due to sparseness in the corpus. Finally, both topicality indicators are significant positive predictors: contexts relevant to a target word are likelier to be nutritious for learning it.

Although AVER.b's results were worse, they are easier to interpret, and differ from the original AVER. Table 3 shows AVER.b linear model's coefficients for each normalized feature. According to this analysis, the features in **boldface** are reliable at $p < .05$ (*) or .0005 (***); one feature is suggestive at $p < .1$ (.).

Table 3: Coefficients of linear model for AVER.b

Feature	Coefficient	
WordBand	***	**0.0508**
Flesch-Kincaid Reading Ease	***	**0.0567**
Flesch-Kincaid Grade	*	**0.0328**
Average word length	*	-0.0199
Unigram logP(t)		0.0052
logP(t w1 w2 w3 w4 | __ w1 w2 w3 w4)	***	**0.0241**
logP(w1 t w2 w3 w4 | w1 __ w2 w3 w4)		-0.0039
logP(w1 w2 t w3 w4 | w1 w2 __ w3 w4)	***	**0.0415**
logP(w1w2w3 t w4 | w1 w2 w3 __ w4)	.	-0.0152
logP(w1 w2 w3 w4 t | w1 w2 w3 w4 __)	***	**0.0321**
Co-occurrence (DISCO S1)	***	**0.0483**
Distributional Similarity (DISCO S2)		0.0031
Intercept	***	**2.5823**

For AVER.b, WordBand is significant and Unigram is not, just the opposite of the original AVER. One reason may be that the AVER.b's context probabilities back off to unigram probability for the 8%-9% of 2-word contexts not listed in the bigram table. Reading Ease, Grade, and Word Length are significant in both models. The five context probabilities show a striking pattern: the first, middle, and last positions in a 5-word context are highly predictive, whereas the other two are not. One possible explanation is that target words tend to be adjacent to function words that provide much less specific information about them. Finally, DISCO S1 was highly significant in both models, but DISCO S2 was significant in the original AVER but not AVER.b. It is not obvious how to explain this difference based on the difference in representation of local context features, i.e., how backoff would steal variance from distributional similarity.

To compare the cross-validation results to a random baseline, Figure 1 shows the ROC for the percentage of good contexts (rated 3 or above) accepted against the percentage of bad (rated below 3) contexts accepted, as the acceptance threshold on the logistic model's output probability varies.

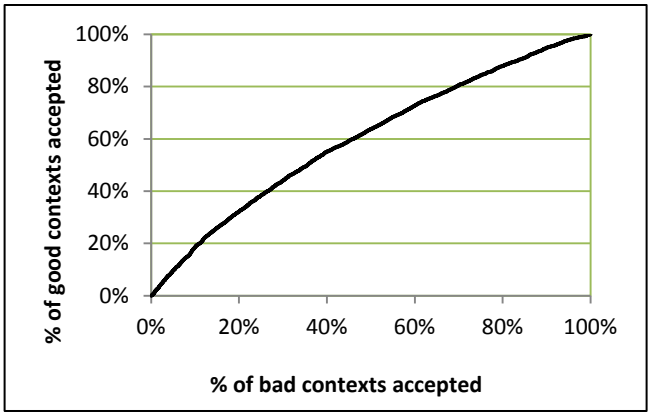

Figure 1: ROC curve for % good vs. % bad contexts accepted

Figure 2 plots the percentages of all the good and bad contexts accepted as the probability threshold decreases from 0.8.

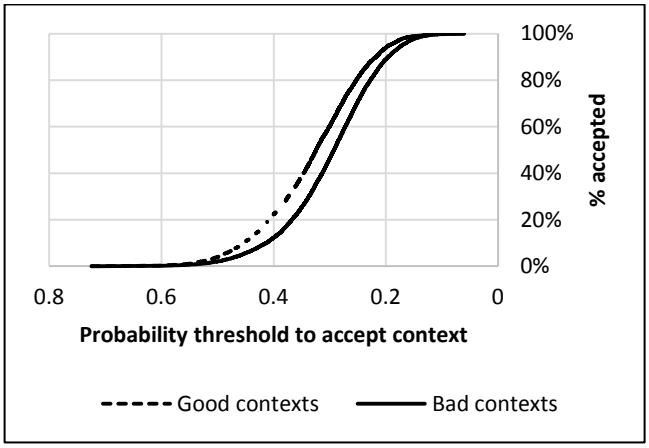

Figure 2: % of contexts accepted vs. probability threshold

As Figure 3 shows, the difference in percentages peaks at 15.2%:

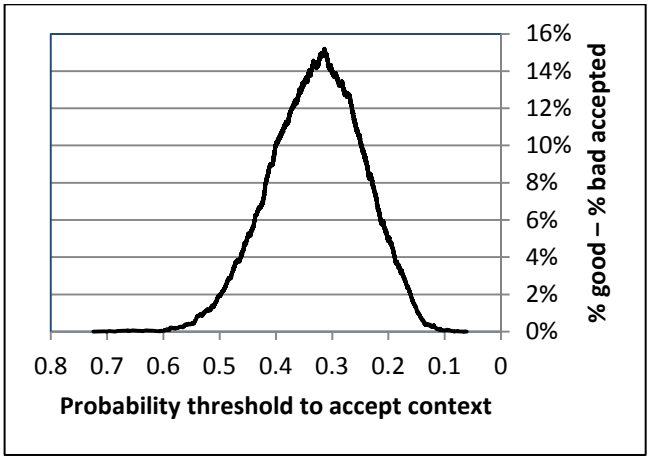

Figure 3: % good – % bad vs. probability threshold

However, bad contexts outnumber good ones, so even when the percentage accepted out of all the good contexts exceeds the percentage accepted out of all the bad contexts, the accepted contexts contains a higher percentage of bad than good contexts, and this imbalance worsens as the threshold decreases, as Figure 4 shows.

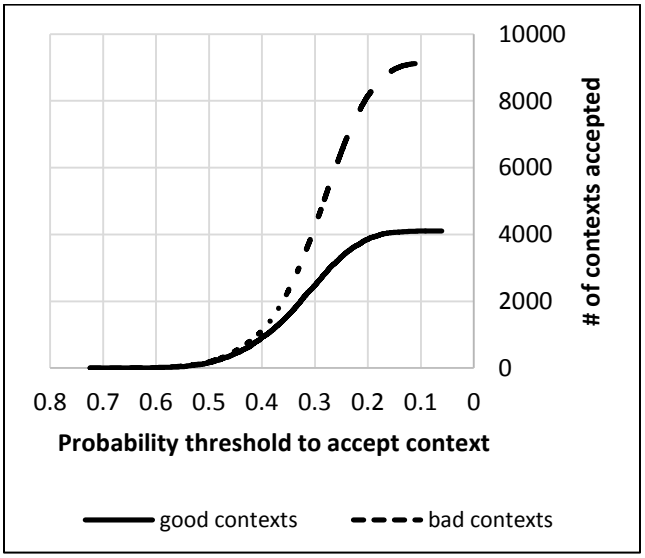

Figure 4: # of contexts accepted vs. probability threshold

As Figure 5 shows, at a threshold of 0.476, the ratio of good to bad contexts reaches a local peak of 0.911 – over twice as high as 0.449, the overall baseline ratio of good contexts to bad contexts. However, at such a high threshold, only 4.4% of the contexts are accepted: 278 (6.8%) of the 4107 good contexts and 305 (3.3%) of the 9150 bad contexts. Thus there is a tradeoff between the number and quality (% good) of the accepted contexts.

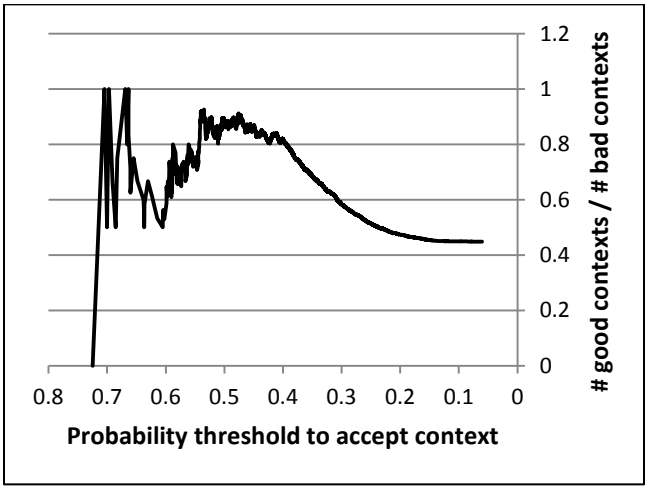

Figure 5: Ratio of good to bad contexts accepted

Visualizing the accuracy of the predicted ratings requires a different type of plot because predicting ratings is not a classification task**.** Accordingly, Figure 6 shows the distribution of errors in rating good and bad contexts as a histogram of predicted minus actual ratings, binned to the nearest 0.1. Figure 6 reflects the fact that there are many more bad than good contexts. It shows that almost all the errors in ratings are less than 1 in size.

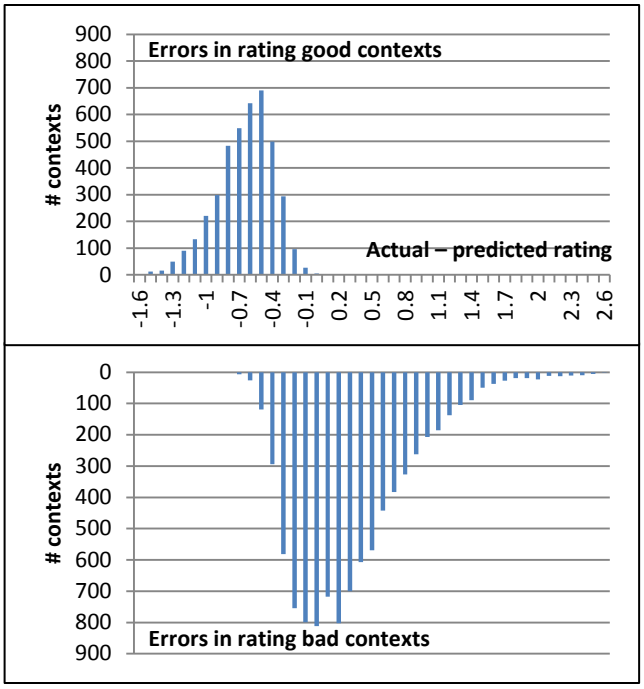

Figure 6: Histogram of errors in rating contexts

7. CONCLUSION

This paper presented and evaluated two models for predicting human ratings of example contexts for learning vocabulary. In contrast to prior work that used manually specified, explicitly operationalized criteria to evaluate contexts, both models approximate the implicit criteria underlying human judgments. Given the wide range of phenomena in language, the diversity of criteria that affect the nutritiousness of example contexts, and humans' limited ability to articulate these criteria explicitly and operationalize them precisely, models trained on human ratings have the potential to surpass hand-crafted models, just as machine learning has surpassed hand-crafted classifiers in other domains.

The AVER system reported here is just an initial step toward this goal: it rates contexts reliably more accurately than chance, but not by very much. Its features are shallow, based on local or bag-of-words statistics rather than deeper linguistic structures such as dependency graphs. Future work should develop more sophisticated features. Our analysis of example contexts with large discrepancies between actual and predicted ratings exposed some promising syntactic and semantic features, informed by human understanding of what makes particular contexts useful to learners or not.

Second, supervised learning from labeled data is only as good as the quality of the labels. The larger project of which this work is a part has already revised the training and selection of raters. However, even expert labels are only a proxy for what actually helps real students. Definitive labels should be grounded empirically in data on how much different students learn about different words from different example contexts. This approach will require considerable amounts of data to be practical – even more so if it tries to model individual differences among students, not just what works well overall on average.

Third, we rated example contexts in isolation, but learning a word's meaning requires encountering it in diverse contexts, not just repeated encounters in the same context, because students

learn different aspects from different contexts. Optimizing the entire sequence of encounters will require identifying what those different aspects are, what sorts of contexts help in learning which aspects, and how learning is affected by their order and how they are related.

Besides accelerating the practical task of selecting good example contexts to teach vocabulary, machine-learned models may eventually shed new light on what properties make example contexts nutritious for learning vocabulary, thereby improving our understanding of human vocabulary learning and instruction.

8. ACKNOWLEDGMENTS

The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A130467 ("Developing an Online Tutor to Accelerate High School Vocabulary Learning") to University of South Carolina (Suzanne Adlof, PI) and its subcontracts to Carnegie Mellon University (Jack Mostow, PI), and University of Pittsburgh (Charles Perfetti, PI). The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education. We thank DictionarySquared founder Adam Kapelner for the contexts, Margaret McKeown for expert ratings, Suzanne Adlof and Julie Byard for Turker ratings, and the reviewers for helpful comments.

9. REFERENCES

[1] Stanovich, K., R. West, and A.E. Cunningham. Beyond phonological processes: Print exposure and orthographic processing**.** In S. Brady and D. Shankweiler, Editors, ***Phonological Processes in Literacy***. Lawrence Erlbaum Associates: Hillsdale, NJ, 1992.

[2] Baumann, J.F., E.J. Kame'enui, and G.E. Ash. Research on vocabulary instruction: Voltaire redux**.** In J. Flood, et al., Editors, ***Handbook of research on teaching the English language arts***, 752-785. Erlbaum & Associates: Mahwah NJ, 2003.

[3] Graves, M.F. Vocabulary learning and instruction**.** In E.Z. Rothkopf, Editor, ***Review of Research in Education***, 91-128 1986.

[4] Mezynski, K. Issues concerning the acquisition of knowledge: Effects of vocabulary training on reading comprehension**.** ***Review of Educational Research***, 1983. ***53***: p. 253-279.

[5] Stahl, S.A. and M.M. Fairbanks. The effects of vocabulary instruction: A model-based meta-analysis**.** ***Review of Educational Research***, 1986. ***56***(1): p. 72-110.

[6] Graves, M.F. A Vocabulary Program to Complement and Bolster a Middle-Grade Comprehension Program**.** In B.M. Taylor, M.F. Graves, and P. van den Broek, Editors, ***Reading for Meaning: Fostering Comprehension in the Middle Grades. Language and Literacy Series***, 116-135. International Reading Association: Newark, DE, 2000.

[7] Biemiller, A. and C. Boote. An effective method for building meaning vocabulary in primary grades**.** ***Journal of Educational Psychology***, 2006. ***98***(1): p. 44-62.

[8] Stahl, S.A. and W.E. Nagy. ***Teaching Word Meanings***. Literacy Teaching Series. 2006, Mahwah, NJ: Lawrence Erlbaum Associates. ix+220.

[9] Beck, I.L., M.G. McKeown, and L. Kucan. ***Bringing Words to Life: Robust Vocabulary Instruction***. 2002, NY: Guilford.

[10] Pavlik Jr., P.I. and J.R. Anderson. Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect**.** ***Cognitive Science***, 2005. ***29***(4): p. 559-586.

[11] Aist, G.S. Helping Children Learn Vocabulary During Computer-Assisted Oral Reading**.** ***Educational Technology and Society***, 2002. ***5***(2): p. http://ifets.ieee.org/periodical/vol_2_2002/aist.html.

[12] Reinking, D. and S.S. Rickman. The effects of computer-mediated texts on the vocabulary learning and comprehension of intermediate-grade readers**.** ***Journal of Reading Behavior***, 1990. ***22***(4).

[13] Brown, J. and M. Eskenazi. Retrieval of Authentic Documents for Reader-Specific Lexical Practice. ***Proceedings of InSTIL/ICALL 2004 Symposium on Computer Assisted Learning***, paper 006. 2004. Venice, Italy.

[14] Liu, L., J. Mostow, and G.S. Aist. Generating Example Contexts to Help Children Learn Word Meaning**.** ***Journal of Natural Language Engineering***, 2013. ***19***(2): p. 187-212.

[15] Mostow, J. and W. Duan. Generating Example Contexts to Illustrate a Target Word Sense**.** In ***Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications***, J. Tetreault, J. Burstein, and C. Leacock, Editors. 2011, Association for Computational Linguistics, Stroudsburg, PA: Portland, OR, p. 105-110. At http://aclweb.org/anthology-new/W/W11/W11-14.pdf.

[16] Franz, A. and T. Brants. All Our *N-gram* are Belong to You**.** 2006. At http://googleresearch.blogspot.com/2006/08/all-our-*n-gram*-are-belong-to-you.html.

[17] Kolb, P. Disco: A multilingual database of distributionally similar words. ***Proceedings of KONVENS-2008 (Konferenz zur Verarbeitung natürlicher Sprache)*** 2008. Berlin.

[18] BNC Consortium. The British National Corpus, version 3 (BNC XML Edition)**.** 2007, Oxford University Computing Services. At http://www.natcorp.ox.ac.uk/.

[19] Surdeanu, M., J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. ***Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations***, 55-60. 2014. Baltimore, MD.

[20] Goldberg, Y. and J. Orwant. A dataset of syntactic-ngrams over time from a very large corpus of english books. ***Second Joint Conference on Lexical and Computational Semantics (* SEM)***, 241-247. 2013.

[21] Jang, H. and J. Mostow. Inferring Selectional Preferences from Part-of-Speech *N-grams*. In ***Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics***. 2012: Avignon, France, p. 377-386.