Analyzing Learner Understanding of Novel L2 Vocabulary

Rebecca Knowles and **Adithya Renduchintala** and **Philipp Koehn** and **Jason Eisner**
Department of Computer Science
Johns Hopkins University
{rknowles,adi.r,phi,eisner}@jhu.edu

Abstract

In this work, we explore how learners can infer second-language noun meanings in the context of their native language. Motivated by an interest in building interactive tools for language learning, we collect data on three word-guessing tasks, analyze their difficulty, and explore the types of errors that novice learners make. We train a log-linear model for predicting our subjects' guesses of word meanings in varying kinds of contexts. The model's predictions correlate well with subject performance, and we provide quantitative and qualitative analyses of both human and model performance.

1 Introduction

Second language (L2) instruction includes an emphasis on vocabulary, as reflected in curricular materials and educational technology. Learners acquire new vocabulary in several ways, including direct instruction, memorization, and incidental acquisition. In this work, we seek a predictive model of the circumstances in which incidental acquisition is possible. That is, when can a learner *guess* the meaning of a novel word?

We present novice learners with new L2 words inserted in sentences otherwise written in their native language (L1). This experimental design allows us to assume that all subjects understand the full context, rather than needing to assess how much of an L2 context each subject understood.

We also present novice learners with the same novel words out of context. This allows us to study how cognateness and context interact, in a well-controlled setting. Cognates and very common words may be easy to translate without context, while contextual clues may be needed to make other words guessable.

In the initial experiments we present here, we focus on the language pair of English L1 and German L2, selecting subjects who self-identify as fluent English speakers with minimal exposure to German. We confine ourselves to novel nouns, as we expect that their relative lack of morphological inflection in both languages[1] will produce less noisy results than verbs, for example. (For verbs, naive learners would be required to attend to tense and mood in addition to the lemma.)

The goal of this work is to develop intuitions that may transfer to less artificial learning settings. Even experienced L2 readers will encounter novel words when reading L2 text. Their ability to decipher a novel word is known to depend on both their understanding of the surrounding context words (to understand a text, a reader needs to understand at least 95% of its words (Huckin and Coady, 1999)) and the cognateness of the novel word. We seek to evaluate this quantitatively and qualitatively in "extreme" cases where the context is either completely comprehensible or absent, and where the cognateness information is either present or absent. In doing so, we are able to see how learners react differently to novel words in different contexts. Our controlled experiments can serve as a proxy for incidental learning in other settings: encountering novel words in isolation (e.g. vocabulary lists), while reading in a familiar language, or while using a language-learning interface such as our own mixed-language reading system (Renduchintala et al., 2016a).

We train a log-linear model to predict the translations that our novice learners will guess, given what we show them and their L1 knowledge. Within this setup, we evaluate the usefulness of a

[1] Both languages mark for number and German occasionally marks for case.

Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pages 126–135,
Berlin, Germany, August 7-12, 2016.

variety of features—that is, we try to identify cues that our learners might plausibly use.

2 Motivation and Related Work

In Renduchintala et al. (2016a) we presented a user interface that allows learners to read "macaronic" (mixed L1/L2) texts, and thus to pick up L2 words and constructions by experiencing them in context. Our interface allows users to click on tokens to translate or reorder words (to make the text more L1-like when they find it too difficult to understand). In the future, we hope to adapt the L1/L2 mix to the individual learner's competence. That is, we wish to present learners with interesting macaronic text that they are able to read with minimal assistance, but which still challenges them: text within the learner's "zone of proximal development" (Vygotsky, 1978).

In order to do this, we must be able to predict when learners will be able to understand a novel L2 vocabulary item. In a previous study (Renduchintala et al., 2016b), we used a small set of simple features to build user-specific models of lexical understanding in macaronic sentences. The present paper evaluates a larger set of features under a more tightly controlled experimental setup. In particular, in the present paper, our model does not have to predict which context words the learner understands, because there is only one L2 word per trial: any context words are always in L1.

A similar project by Labutov and Lipson (2014) likewise considers the effect of context on guessing the L2 word. However, it does not consider the effect of the L2 word's spelling, which we show is also important.

Our experimental setup, particularly the cloze task, is closely related to research in the L2 education and computer-assisted language learning (CALL) domains. Educators often use cloze tasks to evaluate learner vocabulary (though these generally use L2 context). Beinborn et al. (2014a) look at automatically predicting the difficulty of C-tests (a cloze-like task where blanks are introduced at the character level, rather than at the whole-word level). They find features similar to ours to be useful even at the character level, including cognateness, n-gram probabilities, and word length and frequency.

In this work, we focus on predicting the understanding of single words, but this must be extended into larger models of sentence understanding. Vajjala and Meurers (2012) classify the difficulty level of longer L2 texts. Beinborn et al. (2014b) provide an overview of ways that readability measures and user background may be modeled specifically in the context of L2 learners, including through the use of cognateness features. They include a 17-word pilot study of German L1 speakers' ability to guess the meanings of Czech cognates with no context, and hypothesize that observing the words in an understandable context would improve guessability (which we confirm in the English-German case in this work).

3 Data and Methodology

3.1 Textual Data

We use data from `NachrichtenLeicht.de` (Deutschlandfunk, 2016), a source of news articles in Simple German (Leichte Sprache, "easy language"). Simple German is intended for readers with cognitive impairments and/or less than native fluency in German. It follows several guidelines, such as short sentences, simple sentence structure, active voice, hyphenation of compound nouns (which are common in German), and use of prepositions instead of the genitive case (Wikipedia, 2016).

We chose 188 German sentences and manually translated them into English. In each sentence, we selected a single German noun whose translation is a single English noun. This yields a triple of (German noun, English noun, English translation of the context). Each German noun/English noun pair appears only once,[2] for a total of 188 triples. Sentences ranged in length from 5 tokens to 28 tokens, with a mean of 11.47 tokens (median 11). Due to the short length of the sentences, there was often only one possible pair of aligned German and English nouns. In the cases where there were multiple, the translator chose one that had not yet been chosen, and attempted to ensure a wide range of clear cognates to non-cognates, as well as a range of how easy the word was to guess from context.

3.2 Collecting Learner Guesses

Our main goal is to examine learners' ability to understand novel L2 words. To better separate the

[2]The English word may appear in other sentences, but never in the sentence in which its German counterpart appears. In one case, two tuples with different German nouns share the same English noun translation.

Task	Text Presented to Learner	Correct Answer
cloze	The next important _____ conference is in December.	climate
word	*Klima*	climate
combined	The next important *Klima* conference is in December.	climate

Table 1: Three tasks derived from the same German sentence.

effects of context and cognate cues (and general familiarity with the nouns), we assess subjects on the three tasks illustrated in Table 1:

cloze A single noun is deleted from an English sentence, and subjects are asked to fill in the blank.

word Subjects are presented with a single German word out of context, and are asked to provide their best guess for the translation.

combined Subjects are asked to provide their best-guess translation for a single German noun in the context of an English sentence. This is identical to the cloze task, except that the German noun replaces the blank.

We used Amazon Mechanical Turk (henceforth MTurk), a crowdsourcing platform, to recruit subjects and collect their responses to our tasks. Tasks on MTurk are referred to as HITs (Human Intelligence Tasks). In order to qualify for our tasks, subjects completed short surveys on their language skills. They were asked to rate their language proficiency in four languages (English, Spanish, German, and French) on a scale from "None" to "Fluent." The intermediate options were "Up to 1 year of study (or equivalent)" and "More than 1 year of study (or equivalent)".[3] Only subjects who indicated that they were fluent in English but indicated "None" for German experience were permitted to complete the tasks.

Additional stratification of subjects into groups is described in the subsection below. The HITs were presented to subjects in a somewhat randomized order (as per MTurk standard setup).

3.3 Data Collection Protocol

Each triple gives rise to one cloze, one word, and one combined task. For each of those tasks, 9 subjects make guesses, for a total of 27 guesses per triple.

In this setup, each subject may be asked to complete instances of all three tasks. However, the subject is shown at most one task instance derived from a given data triple (for example, at most one line from Table 1). Subjects were paid between \$0.05 and \$0.08 per HIT, where a HIT consists of 5 instances of the same task. Each HIT was completed by 9 unique subjects. Subjects voluntarily completed from 5 to 90 task instances (1–18 hits), with a median of 25 instances (5 HITs). HITs took subjects a median of 80.5 seconds according to the MTurk output timing.

Data was preprocessed to lowercase all guesses and to correct obvious typos.[4] The $188 \times 27 = 5076$ guesses included 1863 unique strings. Of these, 142 were determined to be errors of some sort: 79 were correctable spelling errors, 54 were multiple-word phrases rather than single words, 8 were German words, and 1 was an ambiguous spelling error. In our experiments, we correct obvious typos and then treat all of the other errors as uncorrectable, replacing them with a special out-of-vocabulary token.

3.4 Data Splits

After collecting data on all triples from our subjects, we split the dataset for purposes of predictive modeling. We randomly partitioned the triples into a training set (112 triples), a development set (38 triples), and a test set (38 triples).

Note that the same partition by triples was used across all tasks. As a result, a German noun/English noun pair that appears in test data is genuinely unseen—it did not appear in the training data for *any* task.

4 Modeling Subject Guesses

When developing educational technology, such as a tool for learning vocabulary, we would like a way to compute the difficulty of examples automatically, in order to present learners with an appropri-

[3] Subjects were instructed to list themselves as having experience equivalent to language instruction if they had been exposed to the language by living in a place that it was spoken, playing online language-learning games, or other such experiences, even if they had not studied it in a classroom.

[4] All guesses that were flagged by spell-check were manually checked to see if they constituted typos (e.g., "langauges" for "languages") or spelling errors (e.g., "speach" for "speech") with clear corrections.

ate balance of challenge and guessability. For such an application, it would be useful to know not only whether the learner is likely to correctly guess the vocabulary item, but also whether their incorrect guesses are "close enough" to allow the subject to understand the sentence and proceed with reading. We seek to build models that can predict a subject's likely guesses and their probabilities, given the context with which they have been presented.

We use various features (described below) to characterize and predict subjects' guesses. Feature functions can jointly evaluate a subject's guess with the task instance seen by the subject.

4.1 Guessability and Guess Quality

We train a log-linear model to predict the words that our subjects guess on training data, and we will check its success at this on test data. However, from an engineering perspective, we do not actually need to predict the user's *specific* good or bad answers, but only *whether* they are good or bad. A language-learning interface should display an L2 word only when the user has a good chance of guessing its L1 translation.

Thus we also assess our features and model on the easier task of predicting the *guessability* of a task instance x—that is, the average empirical accuracy of our subjects on this instance, meaning the fraction of the 9 subjects whose guess $\hat{y}$ exactly matched the reference English translation y^*.

Finally, relaxing the exact-match criterion, we evaluate our model's ability to predict the *guess quality*—the average value over subjects of $\mathrm{sim}(\hat{y}, y^*) \in [0, 1]$. Here "sim" denotes Wu-Palmer similarity (Fellbaum, 1998),[5] which is 1 for exact matches, morphological variants (plural/singular), and synonyms; ≈ 0 for antonyms and unrelated words; and intermediate values for words in the same WordNet lexical neighborhood.

4.2 Features

The subject observes a task instance x (consisting of a German word and/or an English context), and guesses an English word $\hat{y}$. We use features of a "candidate" English word y to evaluate whether it is likely to be that guess ($\hat{y} = y$). Our features are functions whose arguments are x and y, and sometimes the true English word y^*. Note that x and y^* are both derived from the triple.

[5]This modifies the definition of guess quality in our previous study (Renduchintala et al., 2016b), where we took "sim" to be the cosine similarity of GloVe embeddings.

The features are divided into three categories according to which properties of x they consider. When a particular feature had several reasonable definitions (e.g., which phonetic representation to use, or whether or not to normalize), we chose—and describe below—the version that correlated most strongly with guessability on training data.

As an outside resource for training language models and other resources consulted by our features, we used Simple English Wikipedia (Wikimedia Foundation, 2016). It contains 767,826 sentences, covers a similar set of topics to the `NachrichtenLeicht.de` data, and uses simple sentence structure. The sentence lengths are also comparable, with a mean of 17.6 tokens and a median of 16 tokens. This makes it well-matched for our task. We also use pre-trained vector representations of words; for these we chose to use the 300-dimensional GloVe vectors trained on a 6B-token dataset by Pennington et al. (2014).

4.2.1 Generic Features

These features ignore x, and hence can be computed in all three tasks.

Log Unigram Frequency of candidate y in the Simple English Wikipedia corpus. A positive weight means that subjects tend to guess more frequent words.

Candidate=Correct Answer This binary feature fires when $y = y^*$. A positive weight on this feature means that subjects are able to guess the correct answer more often than our other features would predict. This may occur because subjects use better features than we do (e.g., their language model analyzes the semantics of the context more deeply than ours) or because they have some outside knowledge of some of the German words, despite not having formally studied German.

Candidate=OOV This binary feature fires when y is not a valid English word (for example, multiple words or an incomprehensible typo), in which case all other features (generic or otherwise) are set to 0.

The following features are "soft" versions of the "Candidate=Correct Answer" feature:

Embedding $1 - \frac{e(y) \cdot e(y^*)}{\|e(y)\|_2 \|e(y^*)\|_2}$ between GloVe embedding of the candidate $e(y)$ and of the correct answer $e(y^*)$.

Levenshtein Distance Unweighted edit distance between y and y^*.

Sound Edit Distance Unweighted edit distance between phonetic representations of y and y^*, as given by Metaphone (Philips, 1990).

LCS Length of longest common substring between y and y^*, normalized by the length of the shorter of the two strings.

Normalized Trigram Overlap count of character trigram types that match between the candidate and correct answer, normalized by the number of trigram types in either the candidate or the correct answer (whichever is smaller).

4.2.2 Word Features

We measure cognateness between the candidate guess y and the German word (which is part of x) using the same 4 string similarity measures used in the final 4 features of the previous section. Note that sound edit distance obtains a pronunciation of the German word using Metaphone, which is designed for English words; this corresponds to the hypothesis that our novice learners may be applying English pronunciation rules to German.

These features depend on the German word, so when used in our models we set them to 0 in the cloze task (where the German word is unobserved).[6]

4.2.3 Cloze Features

The following features depend on the surrounding English context, so they are set 0 in the word task (where the context is unobserved) when used in our models.

Language Model Scores of candidate in context, using a 5-gram language model (LM) built using KenLM (Heafield et al., 2013) and a neural RNN-LM (Mikolov et al., 2011).[7] We compute three different features for each language model: a raw LM score, a sentence-length-normalized LM score, and the difference between the LM score with the correct answer in the sentence and the LM score with the candidate in its place.

PMI Maximum pointwise mutual information between any word in the context and the candidate. This is estimated within a sentence using Simple English Wikipedia and is unsmoothed.

Left Bigram Collocations These are the bigram association measures defined in Church and Hanks (1990) between the candidate's neighbor(s) to the left and the candidate. We train a version that just examines the neighbor directly to the left (which we'd expect to do well in collocations like "San Francisco") as well as one that returns the maximum score over a window of the five previous words.

Context Embeddings The minimum embedding score (defined in 4.2.1) between the candidate and any word in the context.

4.3 Which English Words are Guessable?

Intuitively, we expect it to be hardest to guess the correct English word from the German word alone, followed by guessing it in context, followed by guessing from both cues.[8] As shown in Figure 1, this is borne out in our data.

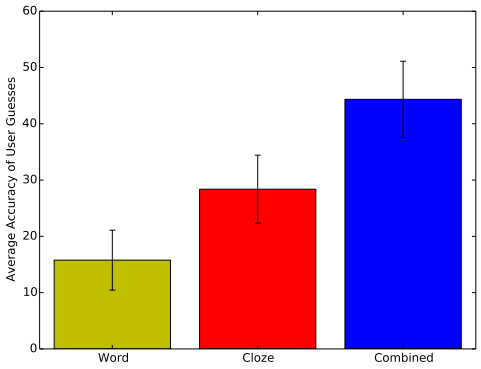

Figure 1: Average guessability (section 4.1) of the 112 training triples, according to which parts of the triple were shown. Error bars show 95%-confidence intervals for the mean, under bootstrap resampling of the 112 triples (we use BCa intervals). Mean accuracy increases significantly from each task to the next (same test on difference of means, $p < 0.01$).

In Table 2 we show Spearman correlations between several features and the guessability of the word (given a word, cloze, or combined context). The first feature in Table 2 (log unigram probability) belongs to the generic category of features. We expect that learners may have an easier time guessing short or common words (for instance, it

[6] In theory, any unavailable features could be indirectly correlated with guessability, but in fact their correlation with guessability is low (absolute value < 0.15) and not statistically significant even at the $p < 0.05$ level.

[7] We use the Faster-RNNLM toolkit available at `https://github.com/yandex/faster-rnnlm`.

[8] All plots/values in the remainder of this section are computed only over the training data unless otherwise noted.

Feature	Correlation w/ Guessability			
	Word	*Cloze*	*Combined*	*All*
Log Unigram Frequency	0.310*	0.262*	0.279*	0.255*
Sound Edit Distance (German + Answer)	-0.633*	n/a	-0.575*	-0.409*
Levenshtein Distance (German + Answer)	-0.606*	n/a	-0.560*	-0.395*
Max PMI (Answer + Context)	n/a	0.480*	0.376*	0.306*
Max Left Bigram Collocations (Answer + Window=5)	n/a	0.474*	0.186	0.238*
Max Right Bigram Collocations (Answer + Window=5)	n/a	0.119	0.064	0.038

Table 2: Spearman's rho correlations between selected feature values and answer guessability, computed on training data (starred correlations significant at $p < 0.01$). Unavailable features are represented by "n/a" (for example, since the German word is not observed in the cloze task, its edit distance to the correct solution is unavailable to the subject).

may be easier to guess *cat* than *trilobite*) and we do observe such correlations.

The middle section focuses on cognateness, which in cases like *Gitarrist* (*guitarist*) can enable all or nearly all subjects to succeed at the challenging word-only task. The correlation between guessability and Sound Edit Distance as well Levenshtein Distance demonstrate their usefulness as proxies for cognateness. The other word features described earlier also show strong correlation with guessability in the word and combined tasks.

Similarly, in some cloze tasks, strong collocations or context clues, as in the case of "His plane landed at the _____." make it easy to guess the correct solution (*airport*). We would expect, for instance, a high PMI between *plane* and *airport*, and we see this reflected in the correlation between high PMI and guessability. The final two lines of the table examine an interesting quirk of bigram association measures. We see that Left Bigram Collocations with a window of 5 (that is, where the feature returns the maximum collocation score between a word in the window to the left of the word to be guessed) shows reasonable correlation with guessability. The reverse, Right Bigram Collocations, however, do not appear to correlate. This suggests that the subjects focus more on the words preceding the blank when formulating their guess (which makes sense as they read left-to-right). Due to its poor performance, we do not include Right Bigram Collocations in our later experiments.

4.4 What English Words are Guessed?

We now move from modeling guessability (via features of the correct answer y^*) to modeling subjects' actual guesses (via features of the guess $\hat{y}$).

We expect that learners who see only the word will make guesses that lean heavily on cognateness (for example, incorrectly guessing *Austria* for *Ausland*), while learners who see the cloze task will choose words that make sense semantically (e.g. incorrectly guessing *tornado* in the sentence "The _____ destroyed many houses and uprooted many trees.").

In Figure 2, we see this holds true; incorrect guesses in the word task have higher average Normalized Character Trigram Overlap than guesses in the cloze task, with the combined task in between. This pattern of the combined task falling between the word and combined task is consistent across most features examined. For example, the difference between the language model scores with the guesses and correct answer is low for the cloze and combined tasks (meaning that users are making guesses that the language model finds about equally plausible to the correct answer), while it is high for the word task (meaning that the users are guessing words that are nonsensical in the context, which they didn't observe). This reinforces that the subjects are making plausible guesses given the cues they observe.

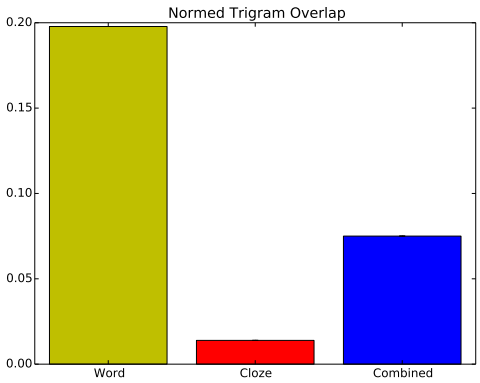

Figure 2: Average Normalized Character Trigram Overlap between incorrect guesses and the German word.

5 Model

The correlations in the previous section support our intuitions about how to model subject behavior in terms of cognateness and context. Section 4.4 suggests that subjects are performing cue combination, balancing cognate and context clues when both are available.

We now build a simple model of cue combination, namely a log-linear model of subjects' guesses:

$$p(y \mid x) = \frac{\exp(\vec{w} \cdot \vec{f}(x, y))}{\sum_{y' \in V} \exp(\vec{w} \cdot \vec{f}(x, y'))} \quad (1)$$

where $\vec{w}$ is a weight vector and $\vec{f}(x, y)$ is a feature vector.

In practice we set V in the denominator to be a 5000-word vocabulary. It contains the complete English vocabulary from the triples (reference translations *and* their context words) as well as all subject guesses. These account for 2238 types (including the special out-of-vocabulary token). To reach 5000 words, we then pad the vocabulary with the most frequent words from the Simple English Wikipedia dataset.

Given the context x that the subject was shown (word, cloze, or combined), $p(y \mid x)$ represents the probability that a subject would guess the vocabulary item $y \in V$. We train the model to maximize the total conditional log-likelihood $\sum_i \log p(\hat{y}_i \mid x_i)$ of all subject guesses $\hat{y}_i$ on all training instances x_i of all three tasks, plus an L2 regularization term.[9]

In order to best leverage the cloze features (shared across the cloze and combined tasks), the word features (shared across the word and combined task) and the generic features (shared across all tasks), we take the domain adaptation approach used in (Daumé III, 2007). In this approach, instead of a single feature for Levenshtein distance between a German word and a candidate guess, we have three copies of this feature, one that fires only when the subject is presented with the word task, one that fires when the subject is presented with the combined task, and a "shared" version that fires in either of those situations. (Note that since a subject who sees the cloze task does not see the German word, we omit such a version of the feature.) This allows us to learn different weights for different tasks. For example, the model can learn that Levenshtein distance is weighted highly in general but especially highly in the word task. The "shared" features mean that the training examples from one task help to set some weights that are used on other tasks (i.e., generalization from limited data), while the task-specific features allow task-specific weights when motivated by the evidence.

5.1 Evaluating the Model

We evaluate the model in several ways: using conditional cross-entropy, by computing mean reciprocal rank, and by examining its ability to predict guessability and guess quality as defined in section 4.1.

The *conditional cross-entropy* is defined to be the mean negative log probability over all test task instances (pairs of subject guesses $\hat{y}$ and contexts x), $\frac{1}{N} \sum_{i=0}^{N} -\log_2 p(\hat{y}_i \mid x_i)$.

The *mean reciprocal rank* is computed after ranking all vocabulary words (in each context) by the probability assigned to them by the model, calculating the reciprocal rank of the each subject guess $\hat{y}_i$, and then averaging this across all contexts x in the set X of all contexts, as shown in Equation 2.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}(\hat{y}_i | x_i)} \quad (2)$$

The model predicts the guessability of x_i to be $p(y_i^* \mid x_i)$, the predicted probability that a user will guess the truth. It predicts the guess quality of x_i, in expectation, to be $\sum_{y \in V} p(y \mid x_i) \, \text{sim}(y, y_i^*)$. We measure how well the *predicted* guessability and guess quality correlate with their *actual* empirical values, using Spearman's rho.[10]

6 Results and Analysis

In Table 3 we show the performance of our full model (last line), as well as several ablated models that use only a subset of the features. The full model performs best. Indeed, an ablated model that uses only generic features, word features, or cloze features cannot reasonably be expected to perform well on the full test set, which contains instances of all three tasks. Using domain adaptation improves performance.

[9] We used MegaM (Daumé III, 2004) via the NLTK interface, with default settings.

[10] In our previous study (Renduchintala et al., 2016b), we measured similar correlations using Pearson's r.

Features	Cross-Entropy	MRR	Guessability Correlation
LCS (Candidate + Answer)	10.72	0.067	0.346*
All Generic Features	8.643	0.309	0.168
Sound Edit Dist. (Cand. + German Word)	10.847	0.081	0.494*
All Word Features	10.018	0.187	0.570*
LM Difference	11.214	0.051	0.398*
All Cloze Features	10.008	0.105	0.351*
Generic + Word	7.651	0.369	0.585*
Generic + Cloze	8.075	0.320	0.264*
Word + Cloze	8.369	0.227	0.706*
All Features (No Domain Adapt.)	7.344	0.338	0.702*
All Features + Domain Adapt.	**7.134**	**0.382**	**0.725***

Table 3: Feature ablation. The single highest-correlating feature (on dev set) from each feature group is shown, followed by the entire feature group. All versions with more than one feature include a feature for the OOV guess. In the correlation column, p-values < 0.01 are marked with an asterisk.

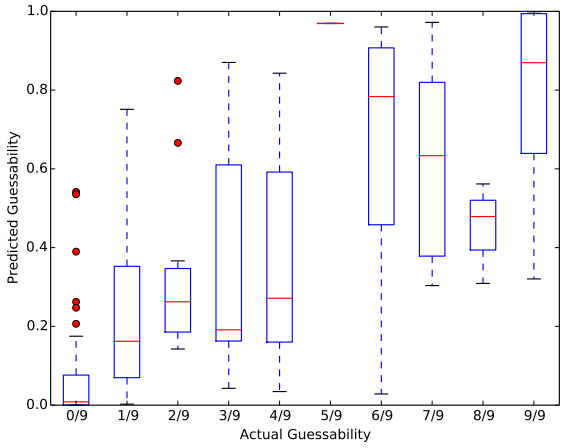

Figure 3: Correlation between actual guessability and the model's prediction of it, across all tasks in the test set. Each point is a task instance, with actual guessability being average $\text{equal}(\hat{y}, y^*) \in \{0, 1\}$ over 9 subjects. Spearman's rank correlation of 0.725.

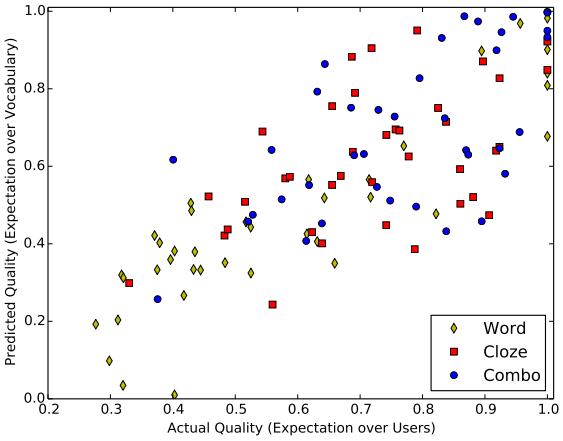

Figure 4: Correlation between actual guess quality and the model's prediction of it. Each point is a task instance, with actual guess quality being average $\text{sim}(\hat{y}, y^*) \in [0, 1]$ over 9 subjects. Spearman's rank correlation of 0.769.

Figure 3 visualizes the correlation shown in our full model (last row of Table 3). This figure illustrates that a single model works well for all three tasks. As the empirical guessability increases, so does the median model probability assigned to the correct answer. However, in our applications, we are less interested in only the 1-best prediction; we'd like to know whether users can understand the novel vocabulary, so we'd prefer to allow WordNet synonyms to also be counted as correct. In Figure 4 we show that the model's prediction of guess quality (see section 4.1) correlates strongly with the actual empirical guess quality.

This means that our model makes predictions that look plausibly like those made by the human subjects. For example, given the context "In _____, the AKP now has the most representatives." the model ranks the correct answer (*parliament*) first, followed by *undersecretary*, *elections*, and *congress*, all of which are thematically appropriate, and most of which fit contextually into the sentence. For the German word *Spieler*, the top ranking predictions made by the model are *spider*, *smaller*, and *spill*, while one of the actual subject guesses, *speaker*, is ranked as 10th most likely (out of a vocabulary of 5000 items).

6.1 Annotated Guesses

To take a fine-grained look at guesses, we broke down subject guesses into several categories.

We had 4 annotators (fluent English speakers,

Context Observed	Guess	Truth	Hypothesized Explanation
Helfer	cow	helpers	False Friend: Helfer → Heifer → Cow
Journalisten	reporter	journalists	Synonym and incorrect number.
The *Lage* is too dangerous.	lake	location	Influenced by spelling and context.

Table 4: Examples of incorrect guesses and potential sources of confusion.

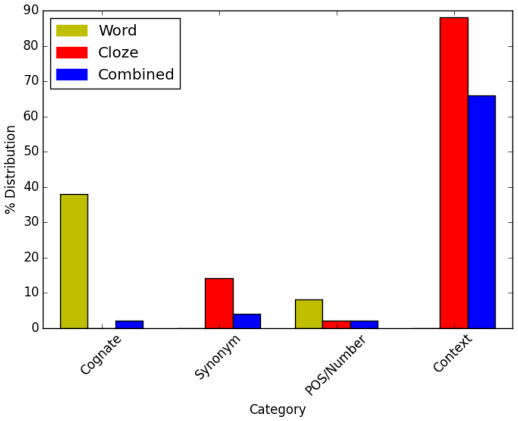

Figure 5: Percent of examples labeled with each label by a majority of annotators (may sum to more than 100%, as multiple labels were allowed).

but non-experts) label 50 incorrect subject guesses from each task, sampled randomly from the spell-corrected incorrect guesses in the training data, with the following labels indicating why the annotator thought the subject made the (incorrect) guess they did, given the context that the subject saw: **false friend/cognate/spelling bias** (learner appears to have been influenced by the spelling of the German word), **synonym** (learner guess is a synonym or near-synonym to the correct answer), **incorrect number/POS** (correct noun with incorrect number or incorrect POS), and **context influence** (a word that makes sense in the cloze/combo context but is not correct). Examples of the range of ways in which errors can manifest are shown in Table 4. Annotators made a binary judgment for each of these labels. Inter-annotator agreement was substantial, with Fleiss's kappa of 0.654. Guesses were given a label only if the majority of annotators agreed.

In Figure 5, we can make several observations about subject behavior. First, the labels for the combined and cloze tasks tend to be more similar to one another, and quite different from the word task labels. In particular, in the majority of cases, subjects completing cloze and combo tasks choose words that fit the context they've observed, while spelling influence in the word task doesn't appear to be quite as strong. Even if the subjects in the cloze and combined tasks make errors, they choose words that still make sense in context more than 50% of the time, while spelling doesn't exert an equally strong influence in the word task.

7 Conclusions

We have shown that by cue combination of various cognate and context features, we can model the behavior of subjects guessing the meanings of novel L2 vocabulary items. Not only does our model correlate well with the guessability of novel words in a variety of contexts, it also produces reasonable predictions for the range of incorrect guesses that subjects make. Such predictions can be used in downstream tasks, such as personalized language learning software, or evaluating the difficulty level of texts.

Acknowledgments

This material is based upon work supported by a National Science Foundation Graduate Research Fellowship (Grant No. DGE-1232825) to the first author and by a seed grant from the Science of Learning Institute at Johns Hopkins University. We thank Chadia Abras for useful discussions, and Nancy Fink, Biman Gujral, Huda Khayrallah and Nitisha Rastogi for volunteering to assist with annotation. We thank the reviewers for their comments and suggestions.

References

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

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014b. Readability for foreign language learning: The importance of cognates. *ITL-International Journal of Applied Linguistics*, 165(2):136–162.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March.

Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. August.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of ACL*, Prague, Czech Republic.

Deutschlandfunk. 2016. Nachrichtenleicht. [Online; accessed 16-March-2016].

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of ACL*, pages 690–696, Sofia, Bulgaria, August.

Thomas Huckin and James Coady. 1999. Incidental vocabulary acquisition in a second language. *Studies in Second Language Acquisition*, 21(02):181–193.

Igor Labutov and Hod Lipson. 2014. Generating code-switched text for lexical learning. In *Proceeding of ACL*, pages 562–571.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. 2011. RNNLM—Recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Lawrence Philips. 1990. Hanging on the metaphone. *Computer Language*, 7(12).

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016a. Creating interactive macaronic interfaces for language learning. In *Proceedings of ACL (System Demonstrations)*.

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016b. User modeling in language learning with macaronic texts. In *Proceedings of ACL*.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173. ACL.

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

Wikimedia Foundation. 2016. Simple English Wikipedia. Retrieved from `https://dumps.wikimedia.org/simplewiki/20160407/` 8-April-2016.

Wikipedia. 2016. Leichte sprache — Wikipedia, die freie enzyklopädie. [Online; accessed 16-March-2016].