Spelling-Aware Construction of Macaronic Texts for Teaching Foreign-Language Vocabulary

Adithya Renduchintala and **Philipp Koehn** and **Jason Eisner**
Center for Language and Speech Processing
Johns Hopkins University
{adi.r,phi}@jhu.edu jason@cs.jhu.edu

Abstract

We present a machine foreign-language teacher that modifies text in a student's native language (L1) by replacing some word tokens with glosses in a foreign language (L2), in such a way that the student can acquire L2 vocabulary simply by reading the resulting *macaronic* text. The machine teacher uses no supervised data from human students. Instead, to guide the machine teacher's choice of which words to replace, we equip a cloze language model with a training procedure that can incrementally learn representations for novel words, and use this model as a proxy for the word guessing and learning ability of real human students. We use Mechanical Turk to evaluate two variants of the student model: (i) one that generates a representation for a novel word using only surrounding context and (ii) an extension that also uses the spelling of the novel word.

1 Introduction

Reading plays an important role in building our native language (L1) vocabulary (Nation, 2001). While some novel words might require the assistance of a dictionary, a large portion are acquired through *incidental learning* – where a reader, exposed to a novel word, tries to infer its meaning using clues from the surrounding context and spelling (Krashen, 1989). An initial "rough" understanding of a novel word might suffice for the reader to continue reading, with subsequent exposures refining their understanding of the novel word.

Our goal is to design a machine teacher that uses a human reader's incidental learning ability to teach foreign language (L2) vocabulary. The machine teacher's modus operandi is to replace L1 words with their L2 glosses, which results in a *macaronic* document that mixes two languages in an effort to ease the human reader into understanding the L2. While some of our prior work (Renduchintala et al., 2016b,a) considered incorporating other features of the L2 such as word order and fixed phrases, in this paper we only consider simple lexical substitutions.

Our hope is that such a system can augment traditional foreign language instruction. As an example, consider a native speaker of English (learning German) presented with the following sentence: **Der** Nile is a **Fluss** in **Afrika**. With a little effort, one would hope the student could infer the meaning of the German words because there is sufficient contextual information and spelling information for the cognate **Afrika**.

In our previous papers on foreign language teaching (Renduchintala et al., 2016b; Knowles et al., 2016; Renduchintala et al., 2017), we focused on fitting detailed models of students' learning when the instructional stimuli (macaronic or otherwise) were chosen by a simple random or heuristic teaching policy. In the present paper, we flip the emphasis to choosing *good* instructional stimuli—machine teaching. This still requires a model of student learning. We employ a reasonable model that is not trained on any human students at all, but only on text that a generic student is presumed to have read. Thus, our model is not personalized, although it may be specialized to the domain of L1 text that it was initially trained on.

That said, our model is reasonably sophisticated and includes new elements. It uses a neural cloze language model (in contrast to the weaker pairwise CRF model of Renduchintala et al. (2016b)) to intelligently guess the meaning of L2 words in full macaronic sentential context. Guessing actually takes the form of a learning rule that jointly improves the embeddings of all L2 words in the sentence. This is our simulation of incidental learning which accumulates over repeated exposures to the same L2 words in different contexts.

Our machine teacher tries to construct macaronic sentences that the human student ought to understand, given all the learning that our generic model predicts would have happened from the previous

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 6438–6443,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

Sentence	The	Nile	is	a	river	in	Africa
Gloss	**Der**	**Nil**	**ist**	**ein**	**Fluss**	**in**	**Afrika**
Macaronic Configurations	**Der**	Nile	**ist**	a	river	in	Africa
	The	Nile	is	a	**Fluss**	in	Africa
	Der	**Nil**	**ist**	**ein**	river	in	Africa

Table 1: An example English (L1) sentence with German (L2) glosses. Using the glosses, many possible macaronic configurations are possible. Note that the gloss sequence is not a fluent L2 sentence.

macaronic sentences shown to the student. Our teacher does not yet attempt to monitor the human student's *actual* learning. Still, we show that it is useful to a beginner student and far less frustrating than a random (or heuristic based) alternative.

A "pilot" version of the present paper appeared at a recent workshop (Renduchintala et al., 2019): it experimented with three variants of the generic student model, using an artificial L2 language. In this paper, we extend the best of those models to consider an L2 word's spelling (along with its context) when guessing its embeddings. We therefore conduct our experiments on real L2 languages (Spanish and German).

2 Related Work

Our motivation is similar to that of commercially available prior systems such as Swych (2015) and OneThirdStories (2018) that also incorporate incidental learning within foreign language instruction. Other prior work (Labutov and Lipson, 2014; Renduchintala et al., 2016b) relied on building a model of the student's incidental learning capabilities, using supervised data that was painfully collected by asking students to react to the actions of an initially untrained machine teacher. Our method, by contrast, constructs a generic student model from unannotated L1 text alone. This makes it possible for us to quickly create macaronic documents in any domain covered by that text corpus.

3 Method

Our machine teacher can be viewed as a search algorithm that tries to find the (approximately) best macaronic configuration for the next sentence in a given L1 document. We assume the availability of a "gold" L2 gloss for each L1 word: in our experiments, we obtained these from bilingual speakers using Mechanical Turk. Table 1 shows an example English sentence with German glosses and three possible macaronic configurations (there are exponentially many configurations). The machine teacher must assess, for example, how accurately a student would understand the meanings of **Der**, **ist**, **ein**, and **Fluss** when presented with the following candidate macaronic configuration: **Der** Nile **ist ein Fluss** in Africa.[1] Understanding may arise from inference on this sentence as well as whatever the student has learned about these words from previous sentences. The teacher makes this assessment by presenting this sentence to a generic student model (§§3.1–3.3). It uses a L2 embedding scoring scheme (§3.4) to guide a greedy search for the best macaronic configuration (§3.5).

3.1 Generic Student Model

Our model of a "generic student" (GSM) is equipped with a cloze language model that uses a bidirectional LSTM to predict L1 words in L1 context (Mousa and Schuller, 2017; Hochreiter and Schmidhuber, 1997). Given a sentence $\mathbf{x} = [x_1, \ldots, x_t, \ldots, x_T]$, the cloze model defines $p(x_t \mid \mathbf{h}^{\mathbf{f}}{}_t, \mathbf{h}^{\mathbf{b}}{}_t) \forall t \in \{1, \ldots, T\}$, where:

$$\mathbf{h}^{\mathbf{f}}{}_t = \text{LSTM}^{\mathbf{f}}([\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}]; \boldsymbol{\theta}^{\mathbf{f}}) \in \mathbb{R}^D \quad (1)$$

$$\mathbf{h}^{\mathbf{b}}{}_t = \text{LSTM}^{\mathbf{b}}([\mathbf{x}_T, \ldots, \mathbf{x}_{t+1}]; \boldsymbol{\theta}^{\mathbf{b}}) \in \mathbb{R}^D \quad (2)$$

are hidden states of forward and backward LSTM encoders parameterized by $\boldsymbol{\theta}^{\mathbf{f}}$ and $\boldsymbol{\theta}^{\mathbf{b}}$ respectively. The model assumes a fixed L1 vocabulary of size V, and the vectors $\mathbf{x}_t$ above are embeddings of these word types, which correspond to the rows of an embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times D}$. The cloze distribution at each position t in the sentence is obtained using

$$p(\cdot \mid \mathbf{h}^{\mathbf{f}}, \mathbf{h}^{\mathbf{b}}) = \text{softmax}(\mathbf{E}\, h([\mathbf{h}^{\mathbf{f}}; \mathbf{h}^{\mathbf{b}}]; \boldsymbol{\theta}^{\mathbf{h}})) \quad (3)$$

where $h(\cdot; \boldsymbol{\theta}^{\mathbf{h}})$ is a projection function that reduces the dimension of the concatenated hidden states from $2D$ to D. We "tie" the input embeddings and output embeddings as in Press and Wolf (2017).

We train the parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}^{\mathbf{f}}; \boldsymbol{\theta}^{\mathbf{b}}; \boldsymbol{\theta}^{\mathbf{h}}; \mathbf{E}]$ using Adam (Kingma and Ba, 2014) to maximize $\sum_{\mathbf{x}} \mathcal{L}(\mathbf{x})$, where the summation is over sentences $\mathbf{x}$ in a large L1 training corpus, and

$$\mathcal{L}(\mathbf{x}) = \sum_t \log p(x_t \mid \mathbf{h}^{\mathbf{f}}{}_t, \mathbf{h}^{\mathbf{b}}{}_t) \quad (4)$$

We set the dimensionality of word embeddings and LSTM hidden units to 300. We use the WikiText-103 corpus (Merity et al., 2016) as the L1 training corpus. We apply dropout ($p = 0.2$) between the word embeddings and LSTM layers, and between the LSTM and projection layers (Srivastava et al., 2014). We assume that the resulting model represents the entirety of the student's L1 knowledge.

[1] By "meaning" we mean the L1 token that was originally in the sentence before it was replaced by an L2 gloss.

3.2 Incremental L2 Vocabulary Learning

Our generic student model (GSM) supposes that to learn new vocabulary, the student continues to try to improve $\mathcal{L}(\mathbf{x})$ on additional sentences. Thus, if x_i is a new word, the student will try to adjust its embedding to increase all summands of (4), both the $t=i$ summand (making x_i more predictable) and the $t\neq i$ summands (making x_i more predictive of x_t).

For our purposes, we do not update $\boldsymbol{\theta}$ (which includes L1 embeddings), as we assume that the student's L1 knowledge has already converged. For the L2 words, we use another word-embedding matrix, $\mathbf{F}$, initialized to $\mathbf{0}^{V'\times D}$, and modify (3) to consider both the L1 and L2 embeddings:

$$p(\cdot \mid [\mathbf{h^f};\mathbf{h^b}]) = \text{softmax}([\mathbf{E};\mathbf{F}]\cdot h([\mathbf{h^f};\mathbf{h^b}];\boldsymbol{\theta^h}))$$

We also restrict the softmax function here to produce a distribution not over the full bilingual vocabulary of size $|V| + |V'|$, but only over the bilingual vocabulary consisting of the L1 types V together with only the L2 types $v' \subset V'$ that actually appear in the macaronic sentence. (In the above example macaronic sentence, $|v'|=4$.) This restriction prevents the model from updating the embeddings of L2 types that are not visible in the macaronic sentence, on the grounds that students are only going to update the meanings of what they are currently reading (and are not even aware of the entire L2 vocabulary).

We assume that when a student reads a macaronic sentence $\mathbf{x}$, they update (only) $\mathbf{F}$ so as to maximize

$$\mathcal{L}(\mathbf{x}) - \lambda \|\mathbf{F} - \mathbf{F^{prev}}\|^2 \tag{5}$$

As mentioned above, increasing the $\mathcal{L}$ term adjusts $\mathbf{F}$ so that the surrounding context can easily predict each L2 word, and each L2 word can, in turn, easily predict the surrounding context (both L1 and L2). However, the penalty term with coefficient $\lambda > 0$ prevents $\mathbf{F}$ from straying too far from $\mathbf{F^{prev}}$, which represents the value of $\mathbf{F}$ before this sentence was read. This limits the degree to which a *single* sentence influences the update to $\mathbf{F}$. As a result, an L2 word's embedding reflects *all* the past sentences that contained that word, not just the most recent such sentence, although with a bias toward the most recent ones, which is realistic. Given a new sentence $\mathbf{x}$, we (approximately) maximize the objective above using 10 steps of gradient ascent (with step-size of 0.1), which gave good convergence in practice. In principle, λ should be set based on human-subject experiments. In practice, in this paper, we simply took $\lambda = 1$.

3.3 Spelling-Aware Extension

So far, our generic student model ignores the fact that a novel word like **Afrika** is guessable simply by its spelling similarity to Africa. Thus, we augment the generic student model to use character n-grams. In addition to an embedding per word type, we learn embeddings for character n-gram types that appear in our L1 corpus. The row in $\mathbf{E}$ for a word w is now parameterized as:

$$\tilde{\mathbf{E}}\cdot\tilde{\mathbf{w}} + \sum_n \tilde{\mathbf{E}}^{\mathbf{n}}\cdot\tilde{\mathbf{w}}^n \frac{1}{\mathbf{1}\cdot\tilde{\mathbf{w}}^n} \tag{6}$$

where $\tilde{\mathbf{E}}$ is the full-word embedding matrix and $\tilde{\mathbf{w}}$ is a one-hot vector associated with the word type w, $\tilde{\mathbf{E}}^{\mathbf{n}}$ is a character n-gram embedding matrix and $\tilde{\mathbf{w}}^n$ is a *multi*-hot vector associated with all the character n-grams for the word type w. For each n, the summand gives the average embedding of all n-grams in w (where $\mathbf{1}\cdot\tilde{\mathbf{w}}^n$ counts these n-grams). We set n to range from 3 to 4 (see Appendix B). This formulation is similar to previous sub-word based embedding models (Wieting et al., 2016; Bojanowski et al., 2017).

Similarly, the embedding of an L2 word w is parameterized as

$$\tilde{\mathbf{F}}\cdot\tilde{\mathbf{w}} + \sum_n \tilde{\mathbf{F}}^{\mathbf{n}}\cdot\tilde{\mathbf{w}}^n \frac{1}{\mathbf{1}\cdot\tilde{\mathbf{w}}^n} \tag{7}$$

Crucially, we initialize $\tilde{\mathbf{F}}^{\mathbf{n}}$ to $\mu\tilde{\mathbf{E}}^{\mathbf{n}}$ (where $\mu > 0$) so that L2 words can inherit part of their initial embedding from similarly spelled L1 words: $\tilde{\mathbf{F}}^{\mathbf{4}}{}_{\texttt{Afri}} := \mu\tilde{\mathbf{E}}^{\mathbf{4}}{}_{\texttt{Afri}}$.[2] But we allow $\tilde{\mathbf{F}}^{\mathbf{n}}$ to diverge over time in case an n-gram functions differently in the two languages. In the same way, we initialize each row of $\tilde{\mathbf{F}}$ to the corresponding row of $\mu\cdot\tilde{\mathbf{E}}$, if any, and otherwise to $\mathbf{0}$. Our experiments set $\mu = 0.2$ (see Appendix B). We refer to this spelling-aware extension to GSM as sGSM.

3.4 Scoring L2 embeddings

Did the simulated student learn correctly and usefully? Let $\mathcal{P}$ be the "reference set" of all (L1 word, L2 gloss) pairs from *all tokens in the entire document*. We assess the machine teacher's success by how many of these pairs the simulated student has learned. (The student may even succeed on some pairs that it has never been shown, thanks to n-gram clues.) Specifically, we measure the "goodness" of

[2] We set $\mu = 0.2$ based on findings from our hyperparameter search (see Appendix B).

the updated L2 word embedding matrix $\mathbf{F}$. For each pair $p = (e, f) \in \mathcal{P}$, sort all the words in the entire L1 vocabulary according to their cosine similarity to the L2 word f, and let r_p denote the rank of e. For example, if the student had managed to learn a matrix $\mathbf{F}$ whose embedding of f exactly equalled $\mathbf{E}$'s embedding of e, then r_p would be 1. We then compute a mean reciprocal rank (MRR) score of $\mathbf{F}$:

$$\text{MRR}(\mathbf{F}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left(\frac{1}{r_p} \textbf{ if } r_p \leq r_{\text{max}} \textbf{ else } 0 \right) \quad (8)$$

We set $r_{\text{max}} = 4$ based on our pilot study. This threshold has the effect of only giving credit to an embedding of f such that the correct e is in the simulated student's top 4 guesses. As a result, §3.5's machine teacher focuses on introducing L2 tokens whose meaning can be deduced *rather accurately* from their single context (together with any prior exposure to that L2 type). This makes the macaronic text comprehensible for a human student, rather than frustrating to read. In our pilot study we found that r_{max} substantially improved human learning.

3.5 Macaronic Configuration Search

Our current machine teacher produces the macaronic document greedily, one sentence at a time. Actual documents produced are shown in Appendix D.

Let $\mathbf{F}^{\text{prev}}$ be the student model's embedding matrix after the reading the first $n-1$ macaronic sentences. We evaluate a candidate next sentence $\mathbf{x}$ by the score $\text{MRR}(\mathbf{F})$ where $\mathbf{F}$ maximizes (5) and is thus the embedding matrix that the student would arrive at after reading $\mathbf{x}$ as the n^{th} macaronic sentence.

We use best-first search to seek a high-scoring $\mathbf{x}$. A search state is a pair $(i, \mathbf{x})$ where $\mathbf{x}$ is a macaronic configuration (Table 1) whose first i tokens may be either L1 or L2, but whose remaining tokens are still L1. The state's score is obtained by evaluating $\mathbf{x}$ as described above. In the initial state, $i = 0$ and $\mathbf{x}$ is the n^{th} sentence of the original L1 document. The state $(i, \mathbf{x})$ is a final state if $i = |\mathbf{x}|$. Otherwise its two successors are $(i+1, \mathbf{x})$ and $(i+1, \mathbf{x}')$, where $\mathbf{x}'$ is identical to $\mathbf{x}$ except that the $(i+1)^{\text{th}}$ token has been replaced by its L2 gloss. The search algorithm maintains a priority queue of states sorted by score. Initially, this contains only the initial state. A step of the algorithm consists of popping the highest-scoring state and, if it is not final, replacing it by its two successors. The queue is then pruned back to the top 8 states. When the queue becomes empty, the algorithm returns the configuration $\mathbf{x}$ from the highest-scoring final state that was ever popped.

L2	Model	Closed-class	Open-class
Es	random	0.74±0.0126(54)	0.61±0.0134(17)
	GSM	0.72±0.0061(54)	0.70±0.0084(17)
	sGSM	**0.82±0.0038**(41)	**0.80±0.0044**(21)
De	random	0.59±0.0054(34)	0.38±0.0065(13)
	GSM	0.80±0.0033(34)	0.78±0.0056(13)
	sGSM	**0.82±0.0063**(33)	**0.79±0.0062**(14)

Table 2: Average token guess quality ($\tau = 0.6$) in the comprehension experiments. The ± denotes a 95% confidence interval computed via bootstrap resampling of the set of human subjects. The % of L1 tokens replaced with L2 glosses is in parentheses. Appendix C evaluates with other choices of τ.

4 Evaluation

Does our machine teacher generate useful macaronic text? To answer this, we measure whether *human* students (i) comprehend the L2 words in context, and (ii) retain knowledge of those L2 words when they are later seen without context.

We assess (i) by displaying each successive sentence of a macaronic document to a human student and asking them to guess the L1 meaning for each L2 token f in the sentence. For a given machine teacher, all human subjects saw the same macaronic document, and each subject's comprehension score is the average quality of their guesses on all the L2 tokens presented by that teacher. A guess's quality $q \in [0,1]$ is a thresholded cosine similarity between the embeddings[3] of the guessed word $\hat{e}$ and the original L1 word e: $q = \text{cs}(e, \hat{e})$ **if** $\text{cs}(e, \hat{e}) \geq \tau$ **else** 0. Thus, $\hat{e} = e$ obtains $q = 1$ (full credit), while $q = 0$ if the guess is "too far" from the truth (as determined by τ).

To assess (ii), we administer an L2 vocabulary quiz after having human subjects *simply* read a macaronic passage (without any guessing as they are reading). They are then asked to guess the L1 translation of each L2 word type that appeared at least once in the passage. We used the same guess quality metric as in (i).[4] This tests if human subjects naturally learn the meanings of L2 words, in informative contexts, well enough to later translate them out of context. The test requires only short-term retention, since we give the vocabulary quiz immediately after a passage is read.

We compared results on macaronic documents constructed with the generic student model (GSM), its spelling-aware variant (sGSM), and a random

[3] Here we used pretrained word embeddings from Mikolov et al. (2018), in order to measure actual semantic similarity.

[4] If multiple L1 types e were glossed in the document with this L2 type, we generously use the e that maximizes $\text{cs}(e, \hat{e})$.

baseline. In the baseline, tokens to replace are randomly chosen while ensuring that each sentence replaces the same number of tokens as in the GSM document. This ignores context, spelling, and prior exposures as reasons to replace a token.

Our evaluation was aimed at native English (L1) speakers learning Spanish or German (L2). We recruited L2 "students" on Amazon Mechanical Turk (MTurk). They were absolute beginners, selected using a placement test and self-reported L2 ability.

4.1 Comprehension Experiments

We used the first chapter of Jane Austen's "Sense and Sensibility" for Spanish, and the first 60 sentences of Franz Kafka's "Metamorphosis" for German. Bilingual speakers provided the L2 glosses (see Appendix A).

For English-Spanish, 11, 8, and 7 subjects were assigned macaronic documents generated with sGSM, GSM, and the random baseline, respectively. The corresponding numbers for English-German were 12, 7 and 7. A total of 39 subjects were used in these experiments (some subjects did both languages). They were given 3 hours to complete the entire document (average completion time was ≈ 1.5 hours) and were compensated \$10.

Table 2 reports the mean comprehension score over all subjects, broken down into comprehension of function words (closed-class POS) and content words (open-class POS).[5] For Spanish, the sGSM-based teacher replaces *more* content words (but fewer function words), and furthermore the replaced words in both cases are *better understood* on average, which we hope leads to more engagement and more learning. For German, by contrast, the number of words replaced does not increase under sGSM, and comprehension only improves marginally. Both GSM and sGSM do strongly outperform the random baseline. But the sGSM-based teacher only replaces a few additional cognates (**hundert** but not **Mutter**), apparently because English-German cognates do not exhibit large *exact* character n-gram overlap. We hypothesize that character skip n-grams might be more appropriate for English-German.

4.2 Retention Experiments

For retention experiments we used the first 25 sentences of our English-Spanish dataset. New participants were recruited and compensated \$5. Each participant was assigned a macaronic document generated with the sGSM, GSM or random model (20, 18, and 22 participants respectively). As Table 3 shows, sGSM's advantage over GSM on comprehension holds up on retention. On the vocabulary quiz, students correctly translated > 30 of the 71 word types they had seen (Table 8), and more than half when near-synonyms earned partial credit (Table 3).

L2	Model	Closed-class	Open-class
Es	random	$0.47 \pm 0.0058 (60)$	$0.40 \pm 0.0041 (46)$
	GSM	$0.48 \pm 0.0084 (60)$	$0.42 \pm 0.0105 (15)$
	sGSM	$\mathbf{0.52 \pm 0.0054} (47)$	$\mathbf{0.50 \pm 0.0037} (24)$

Table 3: Average type guess quality ($\tau = 0.6$) in the retention experiment. The % of L2 gloss types that were shown in the macaronic document is in parentheses. Appendix C evaluates with other choices of τ.

5 Future Work

We would like to explore different character-based compositions such as Kim et al. (2016) that can potentially generalize better across languages. We would further like to extend our work beyond simple lexical learning to allow learning phrasal translations, word reordering, and morphology.

Beyond that, we envision machine teaching interfaces in which the student reader *interacts* with the macaronic text—advancing through the document, clicking on words for hints, and facing occasional quizzes (Renduchintala et al., 2016b)—and with other educational stimuli. As we began to explore in Renduchintala et al. (2016a, 2017), interactions provide feedback that the machine teacher could use to adjust its model of the student's lexicons (here $\mathbf{E}, \mathbf{F}$), inference (here $\boldsymbol{\theta}^{\mathbf{f}}, \boldsymbol{\theta}^{\mathbf{b}}, \boldsymbol{\theta}^{\mathbf{h}}, \mu$), and learning (here λ). In this context, we are interested in using models that are *student-specific* (to reflect individual learning styles), *stochastic* (since the student's observed behavior may be inconsistent owing to distraction or fatigue), and able to model *forgetting* as well as learning (e.g., Settles and Meeder, 2016).

6 Conclusions

We presented a method to generate macaronic (mixed-language) documents to aid foreign language learners with vocabulary acquisition. Our key idea is to derive a model of student learning from only a cloze language model, which uses both context and spelling features. We find that our model-based teacher generates comprehensible macaronic text that promotes vocabulary learning.

[5] https://universaldependencies.org/u/pos/

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Rebecca Knowles, Adithya Renduchintala, Philipp Koehn, and Jason Eisner. 2016. Analyzing learner understanding of novel L2 vocabulary. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 126–135, Berlin, Germany.

Stephen Krashen. 1989. We acquire vocabulary and spelling by reading: Additional evidence for the input hypothesis. *The Modern Language Journal*, 73(4):440–464.

Igor Labutov and Hod Lipson. 2014. Generating code-switched text for lexical learning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–571, Baltimore, Maryland. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Amr Mousa and Björn Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1023–1032, Valencia, Spain.

Ian S. P. Nation. 2001. *Learning vocabulary in another language*. Ernst Klett Sprachen.

OneThirdStories. 2018. OneThirdStories. `https://onethirdstories.com/`. Accessed: 2019-02-20.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain.

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016a. Creating interactive macaronic interfaces for language learning. In *Proceedings of ACL-2016 System Demonstrations*, pages 133–138, Berlin, Germany.

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016b. User modeling in language learning with macaronic texts. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1859–1869, Berlin, Germany.

Adithya Renduchintala, Philipp Koehn, and Jason Eisner. 2017. Knowledge tracing in sequential learning of inflected vocabulary. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 238–247, Vancouver, Canada.

Adithya Renduchintala, Philipp Koehn, and Jason Eisner. 2019. Simple construction of mixed-language texts for vocabulary learning. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, Florence.

Burr Settles and Brendan Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1848–1858, Berlin, Germany.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Swych. 2015. Swych. `http://swych.it/`. Accessed: 2019-02-20.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas.