Simple Construction of Mixed-Language Texts for Vocabulary Learning

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 takes documents written in a student's native language and detects situations where it can replace words with their foreign glosses such that new foreign vocabulary can be learned simply through reading the resulting mixed-language text. We show that it is possible to design such a machine teacher without any supervised data from (human) students. We accomplish this by modifying a cloze language model to incrementally learn new vocabulary items, and use this language model as a proxy for the word guessing and learning ability of real students. Our machine foreign-language teacher decides which subset of words to replace by consulting this language model.

We evaluate three variants of our student proxy language models through a study on Amazon Mechanical Turk (MTurk). We find that MTurk "students" were able to guess the meanings of foreign words introduced by the machine teacher with high accuracy for both function words as well as content words in two out of the three models. In addition, we show that students are able to retain their knowledge about the foreign words after they finish reading the document.

1 Introduction

Proponents of using extensive reading for language acquisition, such as Krashen (1989), argue that much of language acquisition takes place through *incidental learning* , where a reader infers the meaning of unfamiliar vocabulary or structures using the surrounding (perhaps more familiar) context. Unfortunately, when it comes to learning a foreign language (L2), considerable fluency is required before seeing the benefits of incidental learning. But it may be possible to use a student's native language (L1) fluency to introduce new L2 vocabulary. The student's L1 fluency can provide sufficient "scaffolding" (Wood et al., 1976), which we intend to exploit by finding the "zone of proximal development" (Vygotskiĭ, 2012) in which the learner is able to comprehend the text but only by stretching their L2 capacity.

As an example of such *mixed-language* incidental learning, consider a native speaker of English (learning German) presented with the following sentence: **Der** Nile is a **Fluss** in Africa. With a little effort, one would hope a student can infer the meaning of the German words because there is sufficient contextual information. Perhaps with repeated exposure, the student may eventually learn the German words. Our goal is to create a machine teacher that can detect and exploit situations where incidental learning can occur in narrative text (stories, articles etc.). The machine teacher will take a sentence in the student's native language (L1) and replace certain words with their foreign-language (L2) translations, resulting in a mixed-language sentence. We hope that reading mixed-language documents does not feel like a traditional vocabulary learning drill even though novel L2 words can be picked up over time. We envision our method being used *alongside* traditional foreign-language instruction.

Typically, a machine teacher would require supervised data, meaning data on student behaviors and capabilities (Renduchintala et al., 2016; Labutov and Lipson, 2014). This step is expensive, not only from a data collection point of view, but also from the point of view of students, as they would have to give feedback (i.e. generate labeled data) on the actions of an initially untrained machine teacher. However, our machine teacher requires no supervised data from human students. Instead, it uses a cloze language model trained on corpora from the student's native language as a proxy for a human student. Our machine teacher consults this proxy to guide its construction of mixed-language data. Moreover, we create an evaluation dataset that allows us to determine whether students can actually

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 369–379
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

Sentence	The	Nile	is	a	river	in	Africa
Gloss	**Der**	**Nil**	**ist**	**ein**	**Fluss**	**in**	**Afrika**
Mixed-Lang 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, several possible mixed-language configurations are possible. Note that the glosses do not form fluent L2 sentences.

understand our generated texts and learn from them.

We present three variants of our machine teacher, by varying the underlying language models, and study the differences in the mixed-language documents they generate. We evaluate these systems by asking participants on Amazon Mechanical Turk (MTurk) to read these documents and guess the meanings of L2 words as and when they appear (the participants are expected to use the surrounding words to make their guesses). Furthermore, we select the best performing variant and evaluate if participants can actually *learn* the L2 words by letting participants read a mixed-language passage and give a L2 vocabulary quiz at the *end* of passage, where the L2 words are presented in isolation.

2 Approach

Will a student be able to infer the meaning of the L2 tokens I have introduced? This is the fundamental question that a machine teacher must answer when deciding on which words in an L1 sentence should be replaced with L2 glosses. The machine teacher must decide, for example, if a student would correctly guess the meanings of **Der**, **ist**, **ein**, or **Fluss** when presented with this mixed-language *configuration*: **Der** Nile **ist ein Fluss** in Africa.[1] The machine teacher must also ask the same question of many other possible mixed-language configurations. Table 1 shows an example sentence and three mixed-language configurations from among the exponentially many choices. Our approach assumes a 1-to-1 correspondence (i.e. gloss) is available for each L1 token. Clearly, this is not true in general, so we only focus on mixed-language configurations when 1-to-1 glosses *are* possible. If a particular L1 token does not have a gloss, we only consider configurations where that token is always represented in L1.

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

2.1 Student Proxy Model

Before we address the aforementioned question, we must introduce our student proxy model. Concretely, our student proxy model is a cloze language model that uses bidirectional LSTMs to predicts L1 words from their surrounding context (Mousa and Schuller, 2017; Hochreiter and Schmidhuber, 1997). We refer to it as the cLM (cloze language model). Given a L1 sentence $[x_1, x_2, \ldots, x_T]$, the model defines a distribution $p(x_t \mid [\mathbf{h^f} : \mathbf{h^f}])$ at each position in the sentence. Here, $\mathbf{h^f}$ and $\mathbf{h^b}$ are $D-$dimensional hidden states from forward and backward LSTMs.

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

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

The cLM 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 a matrix $\mathbf{E} \in \mathbb{R}^{V \times D}$. The output distribution (over V word types) is obtained by concatenating the hidden states from the forward and backward LSTMs and projecting the resulting $2D$-dimensional state down to D-dimensions using a projection layer $h(\cdot;\boldsymbol{\theta}^{\mathbf{h}})$. Finally, a softmax operation is performed:

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

Note that the softmax layer also uses the word embedding matrix $\mathbf{E}$ when generating the output distribution (Press and Wolf, 2017). This cloze language model encodes left-and-right contextual dependence rather than the typical sequence dependence of standard (unidirectional) language models.

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.

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

We assume that the resulting model represents the entirety of the student's L1 knowledge, and that the L1 parameters $\boldsymbol{\theta}$ will not change further.

2.2 Incremental L2 Vocabulary Learning

The model so far can assign probability to an L1 sentence such as The Nile is a river in Africa, (using Eq. (4)) but what about a mixed-language sentence such as **Der** Nile **ist ein Fluss** in Africa? To accommodate the

new L2 words, we use another word-embedding matrix, $\mathbf{F} \in \mathbb{R}^{V' \times D}$ and modify Eq 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 above to produce a distribution not over the full bilingual vocabulary of size $|V| + |V'|$, but only over the bilingual vocabulary consisting of the V L1 types together with only the $v' \subset V'$ L2 types that actually appear in the mixed-language sentence $\mathbf{x}$. In the above example mixed-language sentence, $|v'|$ is 4. We initialize $\mathbf{F}$ by drawing its elements IID from Uniform$[-0.01, 0.01]$. Thus, all L2 words initially have random embeddings $[-0.01, 0.01]^{1 \times D}$.

These modifications lets us compute $\mathcal{L}(\mathbf{x})$ for a mixed-language sentence $\mathbf{x}$. We assume that when a human student reads a mixed-language sentence $\mathbf{x}$, they update their L2 parameters $\mathbf{F}$ (but not their L1 parameters $\boldsymbol{\theta}$) to increase $\mathcal{L}(\mathbf{x})$. Specifically, we assume that $\mathbf{F}$ will be updated to maximize

$$\mathcal{L}(\mathbf{x}; \boldsymbol{\theta^f}, \boldsymbol{\theta^b}, \boldsymbol{\theta^h}, \mathbf{E}, \mathbf{F}) - \lambda \|\mathbf{F} - \mathbf{F^{prev}}\|^2 \quad (5)$$

Maximizing Eq. (5) adjusts the embeddings of each L2 word in the sentence so that it is more easily predicted from the other L1/L2 words, and also so that it is more helpful at predicting the other L1/L2 words. Since the rest of the model's parameters do not change, we expect to find an embedding for **`Fluss`** that is similar to the embedding for `river`. However, the regularization term with coefficient $\lambda > 0$ prevents $\mathbf{F}$ from straying too far from from $\mathbf{F^{prev}}$, which represents the value of $\mathbf{F}$ before this sentence was read. This limits the degree to which our simulated student will change their embedding of an L2 word such as **`Fluss`** based on a *single* example. As a result, the embedding of **`Fluss`** reflects *all* of the past sentences that contained **`Fluss`**, although (realistically) with some bias toward the most recent such sentences. We do not currently model spacing effects, i.e., forgetting due to the passage of time.

In principle, λ should be set based on human-subjects experiments, and might differ from human to human. In practice, in this paper, we simply took $\lambda = 1$. We (approximately) maximized the objective above using 5 steps of gradient ascent, which gave good convergence in practice.

2.3 Scoring L2 embeddings

The incremental vocabulary learning procedure (Section 2.2) takes a mixed-language configuration and generates a new L2 word-embedding matrix by applying gradient updates to a previous version of the L2 word-embedding matrix. The new matrix represents the proxy student's L2 knowledge after observing the mixed-language configuration.

Thus, if we can score the new L2 embeddings, we can, in essence, score the mixed-language configuration that generated it. The ability to score configurations affords search (Sections 2.4 and 2.5) for high-scoring configurations. With this motivation, we design a scoring function to measure the "goodness" of L2 word-embeddings, $\mathbf{F}$.

The machine teacher evaluates $\mathbf{F}$ with reference to all correct word-gloss pairs from the *entire document*. For our example sentence, the word pairs are {(`The`, **`Der`**), (`is`,**`ist`**), (`a`,**`ein`**), (`river`,**`Fluss`**)}. But the machine teacher also has access to, for example, {(`water`,**`Wasser`**), (`stream`, **`Fluss`**) ...}, which come from elsewhere in the document. Thus, if $\mathcal{P}$ is the set of word pairs, $\{(x_1, f_1), \ldots (x_{|\mathcal{P}|}, f_{|\mathcal{P}|})\}$, we compute:

$$\tilde{r}_p = R(x_p, \mathbf{cs}(\mathbf{F}_{f_p}, \mathbf{E})) \quad (6)$$

$$r_p = \begin{cases} \tilde{r}_p & \text{if } \tilde{r}_p < r_{\text{max}} \\ \infty & \text{otherwise} \end{cases}$$

$$\text{MRR}(\mathbf{F}, \mathbf{E}, r_{\text{max}}) = \frac{1}{|\mathcal{P}|} \sum_p \frac{1}{r_p} \quad (7)$$

where $\mathbf{cs}(\mathbf{F}_f, \mathbf{E})$ denotes the vector of cosine similarities between the embedding of an L2 word f and the entire L1 vocabulary. $R(x, \mathbf{cs}(\mathbf{E}, \mathbf{F}_f))$ queries the rank of the correct L1 word x that pairs with f. r can take values from 1 to $|V|$, but we use a rank threshold r_{max} and force pairs with a rank worse than r_{max} to ∞. Thus, given a word-gloss pairing $\mathcal{P}$, the current state of the L2 embedding matrix $\mathbf{F}$, and the L1 embedding matrix $\mathbf{E}$, we obtain the Mean Reciprocal Rank (MRR) score in (7).

We can think of the scoring function as a "vocabulary test" in which the proxy student gives (its best) r_{max} guesses for each L2 word type and receives a numerical grade.

2.4 Mixed-Language Configuration Search

So far we have detailed our simulated student that would learn from a mixed-language sentence, and a metric to measure how good the learned L2 embeddings would be. Now the machine teacher only has to search for the best mixed-language configuration of a sentence. As there are exponentially many possible configurations to consider,

exhaustive search is infeasible. We use a simple left-to-right greedy search to approximately find the highest scoring configuration for a given sentence. Algorithm 1 shows the pseudo-code for the search process. The inputs to the search algorithm are the initial L2 word-embeddings matrix $\mathbf{F}^{\mathbf{prev}}$, the scoring function MRR(), and the student proxy model SPM(). The algorithm proceeds left to right, making a binary decision at each token: Should the token be replaced with its L2 gloss or left as is? For the first token, these two decisions result in the two configurations: (i) **Der** `Nile...` and (ii) `The Nile...` These configurations are given to the student proxy model which updates the L2 word embeddings. The scoring function (section 2.3) computes a score for each L2 word-embedding matrix and caches the best configuration (i.e. the configuration associated with the highest scoring L2 word-embedding matrix). If two configurations result in the same MRR score, the number of L2 word types exposed is used to break ties. In Algorithm 1, $\rho(\mathbf{c})$ is the function that counts the number of L2 word types exposed in a configuration $\mathbf{c}$.

Algorithm 1 Mixed-Lang. Config. Search

Require: $\mathbf{x}=[x_1,x_2,...,x_T]$ ▷ L1 tokens
Require: $\mathbf{f}=[f_1,f_2,...,f_T]$ ▷ L2 glosses
Require: $\mathbf{E}$ ▷ L1 emb. matrix
Require: $\mathbf{F}^{\mathbf{prev}}$ ▷ initial L2 emb. matrix
Require: SPM ▷ Student Proxy Model
Require: MRR,$r_{\max}$ ▷ Scoring Func., threshold

function SEARCH($\mathbf{x}$,$\mathbf{f}$,$\mathbf{F}^{\mathbf{prev}}$)
  $\mathbf{c}\leftarrow\mathbf{x}$ ▷ initial configuration is the L1 sentence
  $\mathbf{F}\leftarrow\mathbf{F}^{\mathbf{prev}}$
  $s=\text{MRR}(\mathbf{E},\mathbf{F},r_{\max})$
  for $i=1;i\leq T;i++$ **do**
    $\mathbf{c'}\leftarrow c_1\cdots c_{i-1}\,f_i\,x_{i+1}\cdots x_T$
    $\mathbf{\Phi'}=\text{SPM}(\mathbf{F}^{\mathbf{prev}},\mathbf{c'})$
    $s'=\text{MRR}(\mathbf{E},\mathbf{\Phi'},r_{\max})$
    if $(s',-\rho(\mathbf{c'}))\geq(s,-\rho(\mathbf{c}))$ **then**
      $\mathbf{c}\leftarrow\mathbf{c'},\mathbf{F}\leftarrow\mathbf{F'},s\leftarrow s'$
    end if
  end for
  return $\mathbf{c},\mathbf{F}$ ▷ Mixed-Lang. Config.
end function

2.5 Mixed-Language document creation

Our idea is that a sequence of mixed-language configurations is good if it drives the student proxy model's L2 embeddings toward an MRR score close to 1 (maximum possible). Note that we do *not* change the sentence order (we still want a coherent document), just the mixed-language *configuration* of each sentence. For each sentence in turn, we greedily search over mixed-language configurations using Algorithm 1, then choose the configuration that learns the best $\mathbf{F}$, and proceed to the next sentence with $\mathbf{F}^{\mathbf{prev}}$ now set to this learned $\mathbf{F}$.[2] This process is repeated until the end of the document. The pseudo-code for generating an entire document of mixed-language content is shown in Algorithm 2.

Algorithm 2 Mixed-Lang. Document Gen.

Require: $\mathcal{D}=[(\mathbf{x_1},\mathbf{f_1}),...,(\mathbf{x_N},\mathbf{f_N})]$ ▷ Document
Require: $\mathbf{E}$ ▷ L1 emb. matrix
Require: $\mathbf{F^0}$ ▷ initial L2 emb. matrix

function DOCGEN($\mathcal{D}$,$\mathbf{F^0}$)
  $\mathcal{C}=[]$ ▷ Configuration List
  for $i=1;i\leq N;i++$ **do**
    $\mathbf{x}_i,\mathbf{f_i}=\mathcal{D}[i]$
    $\mathbf{c_i},\mathbf{F_i}=\text{SEARCH}(\mathbf{x_i},\mathbf{f_i},\mathbf{F_{i-1}})$
    $\mathcal{C}\leftarrow\mathcal{C}+[\mathbf{c_i}]$
  end for
  return $\mathcal{C}$ ▷ Mixed-Lang. Document
end function

In summary, our machine teacher is composed of (i) a student proxy model which is a contextual L2 word learning model (Sections 2.1 and 2.2) and (ii) a configuration sequence search algorithm (Sections 2.4 and 2.5), which is guided by (iii) an L2 vocabulary scoring function (Section 2.3). In the next section, we describe two variations for the student proxy models.

3 Variations in Student Proxy Models

We developed two variations for the student proxy model to compare and contrast the mixed-language documents that can be generated.

3.1 Unidirectional Language Model

This variation restricts the bidirectional model (from Section 2.1) to be unidirectional (`uLM`) and follows a standard recurrent neural network (RNN) language model (Mikolov et al., 2010).

$$\log p(\mathbf{x})=\sum_t \log p(x_t\,|\,\mathbf{h^f}_t) \tag{8}$$

$$\mathbf{h^f}_t=\text{LSTM}^{\mathbf{f}}(\mathbf{x}_0,...,\mathbf{x}_{t-1};\boldsymbol{\theta}^{\mathbf{f}}) \tag{9}$$

$$p(\cdot\,|\,\mathbf{h^f})=\text{softmax}(\mathbf{E}\cdot\mathbf{h^f}) \tag{10}$$

Once again, $\mathbf{h^f}\in\mathbb{R}^{D\times 1}$ is the hidden state of the LSTM recurrent network, which is parameterized by $\boldsymbol{\theta}^{\mathbf{f}}$, but unlike the model in Section 2.1, no backward LSTM and no projection function is used.

The same procedure from the bidirectional model is used to update L2 word embeddings (Section 2.2). While this model does not explicitly encode context

[2] For the first sentence, we initialize $\mathbf{F}^{\mathbf{prev}}$ to have values randomly between $[-0.01,0.01]$.

from "future" tokens (i.e. words to the right of x_t) , there is still pressure from right-side tokens $x_{t+t:T}$ because the new embeddings will be adjusted to explain the tokens to the right as well. Fixing all the L1 parameters further strengthens this pressure on L2 embeddings from words to their right.

3.2 Direct Prediction Model

The previous two models variants adjust L2 embeddings using gradient steps to improve the pseudo-likelihood of the presented mixed-language sentences. One drawback of such an approach is computation speed caused by the bottleneck introduced by the softmax operation.

We designed an alternate student prediction model that can "directly" predict the embeddings for words in a sentence using contextual information. We refer to this variation as the *Direct Prediction* (DP) model. Like our previous student proxy models, the DP model also uses bidirectional LSTMs to encode context and an L1 word embedding matrix $\mathbf{E}$. However, the DP model does not attempt to produce a distribution over the output vocabulary; instead it tries to predict a real-valued vector using a feed-forward highway network (Srivastava et al., 2015). The DP model's objective is to minimize the mean square error (MSE) between a predicted word embedding and the *true embedding*. For a time-step t, the predicted word embedding $\hat{\mathbf{x}}_t$, is generated by:

$$\mathbf{h^f}_t = \text{LSTM}^{\mathbf{f}}([\mathbf{x}_1,...,\mathbf{x}_{t-1}];\boldsymbol{\theta}^{\mathbf{f}}) \quad (11)$$

$$\mathbf{h^b}_t = \text{LSTM}^{\mathbf{b}}([\mathbf{x}_{t+1},...,\mathbf{x}_T];\boldsymbol{\theta}^{\mathbf{b}}) \quad (12)$$

$$\hat{\mathbf{x}}_t = \text{FF}([\mathbf{x}_t:\mathbf{h^f}_t:\mathbf{h^b}_t];\boldsymbol{\theta}^{\mathbf{w}}) \quad (13)$$

$$\mathcal{L}(\boldsymbol{\theta}^{\mathbf{f}},\boldsymbol{\theta}^{\mathbf{b}},\boldsymbol{\theta}^{\mathbf{w}}) = \sum_t (\hat{\mathbf{x}}_t - \mathbf{x}_t)^2 \quad (14)$$

where $FF(.;\boldsymbol{\theta}^{\mathbf{w}})$ denotes a feed forward highway network with parameters $\boldsymbol{\theta}^{\mathbf{w}}$. Thus, the DP model training requires that we already have the "true embeddings" for all the L1 words in our corpus. We use pretrained L1 word embeddings from FastText as "true embeddings" (Bojanowski et al., 2017). This leaves the LSTM parameters $\boldsymbol{\theta}^{\mathbf{f}},\boldsymbol{\theta}^{\mathbf{b}}$ and the highway feed-forward network parameters $\boldsymbol{\theta}^{\mathbf{w}}$ to be learned. Equation 14 can be minimized by simply copying the input $\mathbf{x}_t$ as the prediction (ignoring all context). We use *masked training* to prevent the model itself from trivially copying (Devlin et al., 2018). We randomly "mask" 30% of the input embeddings during training. This masking operation replaces the original embedding with either (i) $\mathbf{0}$ vectors, or (ii) vectors of a random word in vocabulary, or (iii) vectors of a "neighboring" word from the vocabulary. [3] The loss, however, is always computed with respect to the correct token embedding.

With the L1 parameters of the DP model trained, we turn to L2 learning. Once again the L2 vocabulary is encoded in $\mathbf{F}$, which is initialized to $\mathbf{0}$ (i.e. before any sentence is observed). Consider the configuration: `The Nile is a` **`Fluss`** `in Africa`. The tokens are converted into a sequence of embeddings: $[\mathbf{x}_0 = \mathbf{E}_{x_0}, ... , \mathbf{x}_t = \mathbf{F}_{f_t},...,\mathbf{x}_T = \mathbf{E}_{x_T}]$. Note that at time-step t the L2 word-embedding matrix is used ($t=4, f_t =$ **`Fluss`** for the example above). A prediction $\hat{\mathbf{x}}_t$ is generated by the model using Equations 11-13. Our hope is that the prediction is a "refined" version of the embedding for the L2 word. The refinement arises from considering the context of the L2 word. If **`Fluss`** was not seen before, $\mathbf{x}_t = \mathbf{F}_{f_t} = \mathbf{0}$, forcing the DP model to only use contextual information. We apply a simple update rule that modifies the L2 embeddings based on the direct predictions:

$$\mathbf{F}_{f_t} \leftarrow (1-\eta)\mathbf{F}_{f_t} + \eta\hat{\mathbf{x}}_t \quad (15)$$

where η controls the interpolation between the old values of a word embedding and the new values which have been predicted based on the current mixed sentence. If there are multiple L2 words in a configuration, say at positions i and j (where $i<j$), we can still follow Eq 11–13. However, to allow the predictions $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ to jointly influence each other, we need to execute multiple prediction iterations.

Concretely, let $\mathbf{X} = [\mathbf{x}_0,...,\mathbf{F}_{f_i},...,\mathbf{F}_{f_j},...,\mathbf{x}_T]$ be the sequence of word embeddings for a mixed-language sentence. The DP model generates predictions $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_0,...,\hat{\mathbf{x}}_i,...,\hat{\mathbf{x}}_j,...,\hat{\mathbf{x}}_T]$. We only use its predictions at time-steps corresponding to L2 tokens since the L2 words are those we want to update (Eq 16).

$$\mathbf{X}^1 = \text{DP}(\mathbf{X}^0)$$

$$\text{Where,} \mathbf{X}^0 = [\mathbf{x}_1,...,\mathbf{F}_{f_i},...,\mathbf{F}_{f_j},...,\mathbf{x}_T]$$

$$\mathbf{X}^1 = [\mathbf{x}_1,...,\hat{\mathbf{x}}^1_i,...,\hat{\mathbf{x}}^1_j,...,\mathbf{x}_T] \quad (16)$$

$$\mathbf{X}^k = \text{DP}(\mathbf{X}^{k-1}) \ \forall 0 \le k < K-1 \quad (17)$$

where $\mathbf{X}^1$ contains predictions at i and j and the original L1 word-embeddings in other positions. We then pass $\mathbf{X}^1$ as input again to the DP model. This is executed for K iterations (Eq 17). With

[3] We precompute 20 neighboring words (based on cosine-similarity) for each word in the vocabulary using FastText embeddings before training.

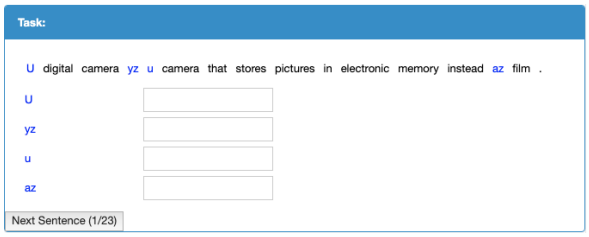

Figure 1: A screenshot of a mixed-language sentence presented on Mechanical Turk.

Metric	Model	$r_{max}=1$	$r_{max}=4$	$r_{max}=8$
Replaced	cLM	0.25	0.31	0.35
	uLM	0.20	0.25	0.25
	DP	0.19	0.22	0.21
Guess Accuracy	cLM	86.00(±0.87)	74.00(±1.10)	55.13(±2.54)
	uLM	84.57(±0.56)	73.89(±1.72)	72.83(±1.58)
	DP	88.44(±0.73)	81.07(±1.03)	70.85(±1.49)

Table 3: Results from MTurk data. The first section shows the percentage of tokens that were replaced with L2 glosses under each condition. The Accuracy section shows the percentage token accuracy of MTurk participants' guesses along with 95% confidence interval calculated via bootstrap resampling.

each iteration, our hope is that the DP model's predictions $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ get refined by influencing each other and result in embeddings that are well-suited to the sentence context. A similar style of imputation has been studied for one dimensional time-series data by Zhou and Huang (2018). Finally, after $K-1$ iterations, we use the predictions of $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ from $\mathbf{X}^K$ to update the L2 word-embeddings in $\mathbf{F}$ corresponding to the L2 tokens f_i and f_j. η was set to 0.3 and the number of iterations $K=5$.

$$\mathbf{F}_{f_i} \leftarrow (1-\eta)\mathbf{F}_{f_i} + \eta\hat{\mathbf{x}}_i^K$$
$$\mathbf{F}_{f_j} \leftarrow (1-\eta)\mathbf{F}_{f_j} + \eta\hat{\mathbf{x}}_j^K \quad (18)$$

4 Experiments

We first investigate the patterns of word replacement produced by the machine teacher under the influence of the different student proxy models and how these replacements affect the guessability of L2 words. To this end, we used the machine teacher to generate mixed-language documents and asked MTurk participants to guess the foreign words. Figure 1 shows an example screenshot of our guessing interface. The words in blue are L2 words whose meaning (in English) is guessed by MTurk participants. For our study, we created a synthetic L2 language by randomly replacing characters from English word types. This step lets us safely assume that all MTurk participants are "absolute beginners." We tried to ensure that the resulting synthetic words are pronounceable by replacing vowels with vowels, stop-consonants with other stop-consonants, etc. We also inserted or deleted one character from some of the words to prevent the reader from using the length of the synthetic word as a clue. While our evaluation required use of a synthetic foreign language, we provide as an example mixed-language documents with real L2 languages in Appendix A.1.

We studied the three student proxy models (cLM , uLM , and DP) while keeping the rest of the machine teacher's components fixed (i.e. same scoring function and search algorithms). All three models were constructed to have roughly the same number of L1 parameters ($\approx 20M$). The uLM model used 2 unidirectional LSTM layers instead of a single bidirectional layer. The L1 and L2 word embedding size and the number of recurrent units D were set to 300 for all three models (to match the size of FastText's pretrained embeddings). We trained the three models on the Wikipedia-103 corpus (Merity et al., 2016).[4] All models were trained for 8 epochs using the Adam optimizer (Kingma and Ba, 2014). We limit the L1 vocabulary to the $60k$ most frequent English types.

4.1 MTurk Setup

We selected 6 documents from Simple Wikipedia to serve as the input for mixed-language content.[5] To keep our study short enough for MTurk, we selected documents that contained $20-25$ sentences. A participant could complete up to 6 HITs (Human Intelligence Tasks) corresponding to the 6 documents. Participants were given 25 minutes to complete each HIT (on average, the participants took 12 minutes to complete the HITs). To prevent typos, we used a $20k$ word English dictionary, which includes all the word types from the 6 Simple Wikipedia documents. We provided no feedback regarding the correctness of guesses. We recruited 128 English speaking MTurk participants and obtained 162 responses, with each response encompassing a participant's guesses over a full document.[6] Participants were compensated $4 per HIT.

4.2 Experiment Conditions

We generated 9 mixed-language versions (3 models {cLM ,uLM ,DP } in combination with 3 rank

[4] FastText pretrained embeddings were trained on more data.
[5] https://dumps.wikimedia.org/simplewiki/20190120/
[6] Participants self-reported their English proficiency, only native or fluent speakers were allowed to participate. Our HITs were only available to participants from the US.

Model	$r_{max}=1$	$r_{max}=8$
cLM	**Hu** Nile (``an-nīl'') **ev** a river **um** Africa. **Up** is **hu** longest river **iñ** Earth (about 6,650 km or 4,132 miles), though other rivers carry more water... Many **ozvolomb** types **iv emoner** live in or near **hu** waters **iv hu** Nile, including crocodiles, birds, fish **ñb** many others. Not only do animals depend **iñ hu** Nile for survival, but also people who live there need **up zi** everyday use like washing, as **u jopi** supply, keeping crops watered **ñb** other jobs...	**Hu** Nile (``an-nīl'') **ev u** river **um** Africa. **Up ev** the longest river on Earth (about 6,650 km or 4,132 miles), though other rivers carry more water... **Emu ozvolomb** types of **emoner** live **um** or **iul** the waters of **hu Uro**, including crocodiles, **ultf**, **yvh** and **emu** others. **Ip** only do animals depend **iñ** the Nile **zi** survival, but also **daudr** who live there need **up zi** everyday use like washing, **ez** a **jopi** supply, keeping crops watered **ñb** other jobs...
uLM	The Nile (``an-nīl'') **ev** a river **um** Africa. It **ev hu** longest river on Earth (about 6,650 km or 4,132 miles), though other rivers carry more **jopi**... Many different **pita** of **emoner** live in or near **hu** waters **iv hu** Nile, including crocodiles, **ultf**, fish and many others. Not **mru** do **emoner** depend **iñ hu** Nile for survival, but also people who live there need it for everyday use like washing, as a **jopi** supply, keeping crops watered **ñb** other jobs...	**Hu** Nile (``an-nīl'') **ev u** river **um** Africa. **Up ev** the longest river **iñ** Earth (about 6,650 km or 4,132 miles), though other rivers carry more **jopi**... Many different **pita** of **emoner** live **um** or near **hu** waters **iv hu** Nile, including crocodiles, **ultf**, fish and many others. Not **mru** do **emoner** depend on the Nile for survival, **id** also people who live there need it **zi** everyday use like washing, as **u** water supply, keeping crops watered **ñb** other jobs...
DP	**Hu** Nile (``an-nīl'') **ev** a river **um** Africa. **Up ev hu** longest river on Earth (about 6,650 km or 4,132 miles), though other rivers carry more water... Many different types **iv** animals live in or near **hu** waters **iv hu** Nile, including crocodiles, birds, fish and many others. Not only do animals depend **iñ hu** Nile for survival, but also people who live there need it for everyday use like washing, as **u** water supply, keeping crops watered and other jobs...	**Hu** Nile (``an-nīl'') **ev** a river **um** Africa. **Up ev hu** longest river on Earth (about 6,650 km or 4,132 miles), though **udho** rivers carry more water... Many different **pita** of animals live in or near **hu** waters of **hu** Nile, including crocodiles, birds, fish and many others. Not **mru** do animals depend **iñ hu** Nile **zi** survival, **id** also people who live there need it **zi** everyday use like washing, **ez** a water supply, keeping crops watered and **udho** jobs...

Table 2: Portions of one of our Simple Wikipedia articles. The document has been converted into a mixed-language document by the machine teacher using the three student proxy models. Our experiments use a synthetic L2 language, see Appendix A.1 for examples with real L2 language (German and Spanish) on two stories. The two columns show the effect of the rank threshold r_{max}. Note that this mixed-language document is 25 sentences long; here, we only show the first 2 sentences and another middle 2 sentences to save space.

thresholds $r_{max} \in \{1,4,8\}$) for each of the 6 Simple Wikipedia documents. For each HIT, an MTurk participant was randomly assigned one of the 9 mixed-language versions. Table 2 shows the output at two settings of r_{max} for one of the documents. We see that r_{max} controls the number of L2 words the machine teacher deems guessable, which affects text readability. The increase in L2 words is most noticeable with the cLM model. We also see that the DP model differs from the others by favoring high frequency words almost exclusively. While the cLM and uLM models similarly replace a number of high frequency words, they also occasionally replace lower frequency word classes like nouns and adjectives (**emoner**, **Emu**, etc.). Table 3 summarizes our findings. The first section of 3 shows the percentage of tokens that were deemed guessable by our machine teacher. The cLM model replaces more words as r_{max} is increased to 8, but we see that MTurkers had a hard time guessing the meaning of the replaced tokens: their guessing accuracy drops to 55% at $r_{max} = 8$ with the cLM model. The uLM model, however, displays a reluctance to replace too many tokens, even as r_{max} was increased to 8.

We further analyzed the replacements and MTurk guesses based on word-class. We tagged the L1 tokens with their part-of-speech and categorized tokens into open or closed class following Universal Dependency guidelines (Nivre et al.).[7] Table 4 summarizes our analysis of model and human behavior when the data is separated by word-class. The pink bars indicate the percentage of tokens replaced per word-class. The blue bars represent the percentage of tokens from a particular word-class that MTurk users *guessed correctly*. Thus, an ideal machine teacher should strive for the highest possible pink bar while ensuring that the blue bar is as close as possible to the pink. Our findings suggest that the uLM model at $r_{max}=8$ and the cLM model at $r_{max} = 4$ show the desirable properties – high guessing accuracy and more representation of L2 words (particularly open-class words).

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

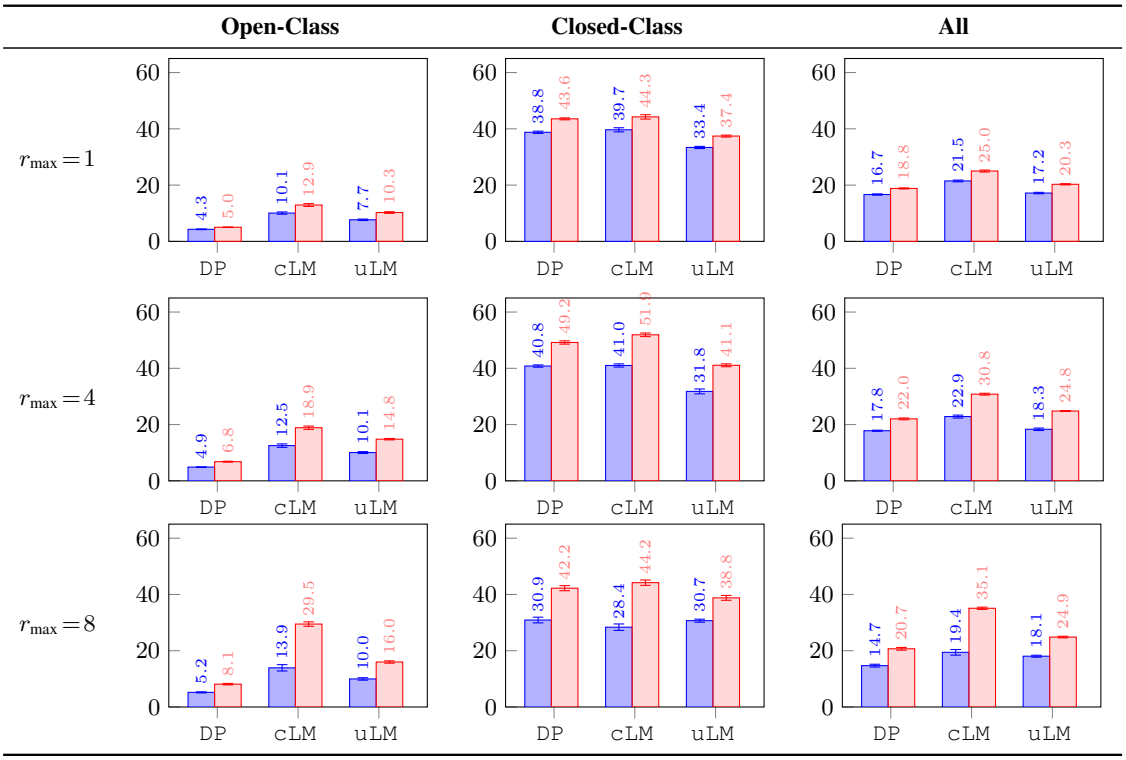

Table 4: Results of MTurk results split up by word-class. The y-axis is percentage of tokens belonging to a word-class. The pink bar (right) shows the percentage of tokens (of a particular word-class) that were *replaced* with an L2 gloss. The blue bar (left) and indicates the percentage of tokens (of a particular word-class) that were *guessed correctly* by MTurk participants. Error bars represent 95% confidence intervals computed with bootstrap resampling. For example, we see that only 5.0% (pink) of open-class tokens were replaced into L2 by the DP model at $r_{max} = 1$ and 4.3% of all open-class tokens were guessed correctly. Thus, even though the guess accuracy for DP at $r_{max} = 1$ for open-class is high (86%) we can see that participants were not exposed to many open-class word tokens.

Metric	Model	Closed	Open
Types Replaced	random	59	524
	cLM	33	149
Guess Accuracy	random	62.06(±1.54)	39.36(±1.75)
	cLM	**74.91(±0.94)**	**61.96(±1.24)**

Table 5: Results comparing our student proxy based approach to a random baseline. The first part shows the number of L2 word types exposed by each model for each word-class. The second part shows the average guess accuracy percentage for each model and word-class. 95% confidence intervals (in brackets) were computed using bootstrap resampling.

4.3 Random Baseline

So far we've compared different student proxy models against each other, but is our student proxy based approach required at all? How much better (or worse) is this approach compared to a *random* baseline? To answer these questions, we compare the cLM with $r_{max} = 4$ model against a randomly generated mixed-language document. As the name suggests, word replacements are decided randomly for the random condition, but we ensure that the number of tokens replaced in each sentence equals that from the cLM condition.

We used the 6 Simple Wikipedia documents from Section 4.1 and recruited 64 new MTurk partipants who completed a total of 66 HITs (compensation was $4 per HIT). For each HIT, the participant was given either the randomly generated or the cLM based mixed-language document. Once again, participants were made to enter their guess for each L2 word that appears in a sentence. The results are summarized in Table 5.

We find that randomly replacing words with glosses exposes more L2 word types (59 and 524 closed-class and open-class words respectively) while the cLM model is more conservative with replacements (33 and 149). However, the random mixed-language document is much harder to comprehend, indicated by significantly lower average guess accuracies than those with the cLM model. This is especially true for open-class words. Note that Table 5 shows the number of word types replaced across all 6 documents.

Model	Closed	Open
random	9.86(±0.94)	4.28(±0.69)
cLM	**35.53(±1.03)**	**27.77(±1.03)**

Table 6: Results of our L2 learning experiments where MTurk subjects simply read a mixed-language document and answered a vocabulary quiz at the end of the passage. The table shows the average guess accuracy percentage along with 95% confidence intervals computed from bootstrap resampling.

4.4 Learning Evaluation

Our mixed-language based approach relies on incidental learning, which states that if a novel word is repeatedly presented to a student with sufficient context, the student will eventually be able to learn the novel word. So far our experiments test MTurk participants on the "guessability" of novel words in context, but not learning. To study if students can actually learn the L2 words, we conduct an MTurk experiment where participants are simply required to read a mixed-language document (one sentence at a time). At the end of the document an L2 vocabulary quiz is given. Participants must enter the meaning of every L2 word type they have seen during the reading phase.

Once again, we compare our cLM ($r_{\max} = 4$) model against a random baseline using the 6 Simple Wikipedia documents. 47 HITs were obtained from 45 MTurk participants for this experiment. Participants were made aware that there would be a vocabulary quiz at the end of the document. Our findings are summarized in Table 6. We find the accuracy of guesses for the vocabulary quiz at the end of the document is considerably lower than guesses with context. However, subjects still managed to retain 35.53% and 27.77% of closed-class and open-class L2 word types respectively. On the other hand, when a random mixed-language document was presented to participants, their guess accuracy dropped to 9.86% and 4.28% for closed and open class words respectively. Thus, even though more word types were exposed by the random baseline, fewer words were retained.

5 Related Work

Our work does not require any supervised data collection from students. This departure makes our work easier to deploy in diverse settings (i.e. for different document genres, and different combinations of L1/L2 languages etc). While there are numerous self-directed language learning applications such as Duolingo (von Ahn, 2013), our approach uses a different style of "instruction". Furthermore, reading L2 words in L1 contexts is also gaining popularity in commercial applications like Swych (2015) and OneThirdStories (2018).

Most recently, Renduchintala et al. (2016) attempt to model a student's ability to guess the meaning of foreign language words (and phrases) when prompted with a mixed language sentence. One drawback of this approach is its need for large amounts of training data, which involves prompting students (in their case, MTurk users) with mixed language sentences created randomly. Such a method is potentially inefficient, as random configurations presented to users (to obtain their guesses) would not reliably match those that a beginner student would encounter. Labutov and Lipson (2014) also use a similar supervised approach. The authors required two sets of annotations, first soliciting guesses of missing words in a sentences and then obtaining another set of annotations to judge the guesses.

6 Conclusion

We are encouraged by the ability to generate mixed-language documents without the need of expensive data collection from students. Our MTurk study shows that students can guess the meaning of foreign words in context with high accuracy and also retain the foreign words.

For future work, we would like to investigate ways to smoothly adapt our student proxy models into personalized models. We also recognize that our approach may be "low-recall," i.e., it might miss out on teaching possibilities. For example, our machine teacher may not realize that cognates can be replaced with the L2 and still understood, even if there are no contextual clues (**Afrika** can likely be understood without much context). Incorporating spelling information into our language models (Kim et al., 2016) could help the machine teacher identify more instances for incidental learning. Additionally, we would like to investigate how our approach could be extended to enable phrasal learning (which should consider word-ordering differences between the L1 and L2). As the cLM and uLM models showed the most promising results in our experiments, we believe these models could serve as the baseline for future work.

References

Luis von Ahn. 2013. Duolingo: Learn a language for free while helping to translate the web. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, pages 1–2.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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*.

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 ACL*, pages 562–571.

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

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Amr El-Desoky Mousa and Björn Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings EACL*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, et al. Universal dependencies v1: A multilingual treebank collection.

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, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 157–163.

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016. 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. Association for Computational Linguistics.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

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

Lev Vygotskiĭ. 2012. *Thought and Language (Revised and Expanded Edition)*. MIT Press.

David Wood, Jerome S. Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100.

Jingguang Zhou and Zili Huang. 2018. Recover missing sensor data with iterative imputing network. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

Sense **y** Sensibility
CHAPTER **1**
La family **de** Dashwood **llevaba** long been settled **en** Sussex. Their estate **era** large, and their residence was **en** Norland Park, **en el** centre **de** their **propiedad**, where, **por** many generations, **ellos** had lived **en** so respectable a manner as **a** engage the general **buena** opinion of their surrounding acquaintance. **El** late owner **de esta** estate was a single man, who lived to **una** very advanced age, and who for many **años de su** life, had **una** constant companion **y** housekeeper in **su** sister. But her death, which happened ten **años** before **su** own, produced a great alteration **en** his home; for **para** supply her loss, he invited **y** received into his house the family of his nephew Mr. **Henry** Dashwood, the legal inheritor **de** the Norland estate, **y** the person to whom **se** intended to bequeath it. **En la** society of his nephew and niece, and their children, **el** old Gentleman's days **fueron** comfortably spent. **Su** attachment **a** them all increased. **La** constant attention **de** Mr. **y** Mrs. Henry Dashwood **a sus** wishes, which proceeded not merely from interest, but from goodness **de** heart, **dio** him every degree **de** solid comfort which his age could receive; **y la** cheerfulness **de los** children added **un** relish to his existence.
Por a former marriage, Mr. **Henry** Dashwood had one **hijo**: by **su** present lady, **tres** daughters. **El** son, **un** steady respectable young man, **tenía** amply provided for by **la** fortune **de su** mother, which **había** been large, **y** half **de** which devolved on him on **su** coming **de** age. **Por** his own marriage, likewise, which happened soon afterwards, he added **a su** wealth. **Para** him therefore **la** succession **a la** Norland estate **era** not so really important **como para** his sisters; **para su** fortune, independent of what might arise **a** them **de su** father's inheriting that **propiedad**, could **ser** but small. **Su madre** had nothing, and their father only seven thousand pounds **en su** own disposal; **porque** the remaining moiety of **su** first wife's fortune **era** also secured **a su** child, **y** he had only a life-interest **en** it.

Table 7: Example of mixed-language output for Jane Austen's "Sense and Sensibility". We used the uLM with $r_{\max} = 8$.

Metamorphosis
I
One morning, when Gregor Samsa woke from troubled dreams, **er** found himself transformed **in** his bed into **einem** horrible vermin. **Er** lay **auf** his armour-like back, **und** if **er** lifted **seinen** head a **wenig** he could see his brown belly, slightly domed **und** divided **von** arches into stiff sections. **das** bedding was hardly able **zu** cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared **mit der** size of **dem** rest of him, waved about helplessly **als** he looked.
''What's happened **mit** me?'' **er** thought. His room, **ein** proper human room although a **wenig** too small, lay peacefully between **seinen** four familiar walls. **Eine** collection of textile samples lay spread out on **dem** table - Samsa was **ein** travelling salesman - **und** above it there hung **ein** picture that **er** had recently cut out **von** an illustrated magazine and housed **in** a nice, gilded frame. It showed **eine** lady fitted out with **einem** fur hat **und** fur boa who sat upright, raising **einen** heavy fur muff that covered the whole of her lower arm towards **dem** viewer.
Gregor **dann** turned to look out the window at the dull weather. Drops of rain could **sein** heard hitting the pane, which **machte** him feel quite sad. ''How about if I sleep **ein** little bit longer and forget all this nonsense,'' **er** thought, but that **war** something **er war** unable **zu** do because he **war** used **zu** sleeping on **seiner** right, **und** in **seinem** present state couldn't get into **diese** position. However hard he threw himself onto **seine** right, **er** always rolled **zurück** to where he was. **Er** must **haben** tried it **ein** hundred times, shut **seine** eyes so **dass er** wouldn't have to look at **die** floundering legs, **und** only stopped when **er** began to feel **einen** mild, dull pain there that **er** had **nie** felt before.
''Oh, God,'' **er** thought, ''what a strenuous career it **ist** that I've chosen! Travelling day in **und** day out. Doing business like **diese** takes much **mehr** effort than doing your own **Geschäft** at home, **und auf** top of that there's **der** curse **des** travelling, worries about making train connections, bad and irregular food, contact with **verschiedenen** people all **die** time so **das** you **kannst** never get to know anyone or become friendly **mit** them. **es** can all **gehen** to Hell!'' **Er** felt a slight itch up **auf seinem** belly ; pushed himself slowly up on **seinen** back towards the headboard so **dass** he **konnte** lift **seinen** head better ; found where **das** itch was, **und** saw **dass** it was **besetzt** with lots of little white spots which **er** didn't know what to make of ; **und** when **er** tried to feel **die** place with one of his legs **er** drew **es** quickly back because as soon as he touched it **er** was overcome by **einem** cold shudder.

Table 8: Example of mixed-language output for the English translation (by David Wyllie) of Franz Kafka's "Metamorphosis". We used the uLM with $r_{\max} = 8$.

A Appendices

A.1 Mixed-Language Examples

While our experiments necessitated use of synthetic L2 words, our methods are compatible with real L2 learning. For a more "real-world" experience of how our methods could be deployed, we present the first few paragraphs of mixed-language novels generated using the uLM model with $r_{\max} = 8$. First example is from Jane Austin's "Sense and Sensibility" (Table 7), and for the second example, as we are transforming text from one language into a "strange hybrid creature" (i.e mixed-language) it seems appropriate to use Franz Kafka's "Metamorphosis"(Table 8). For these examples, glosses were obtained from a previous MTurk data collection process from bilingual speakers. Glosses for each English (L1) token was obtained from 3 MTurkers, if a majority of them agree on the gloss it is considered by our machine teacher as a possible L2 gloss. If no agreement was obtained we restrict that token to always remain as L1.