Learning to Discover, Ground and Use Words with Segmental Neural Language Models

Kazuya Kawakami[♠♣] **Chris Dyer**[♣] **Phil Blunsom**[♠♣]
[♠]Department of Computer Science, University of Oxford, Oxford, UK
[♣]DeepMind, London, UK
{kawakamik,cdyer,pblunsom}@google.com

Abstract

We propose a segmental neural language model that combines the generalization power of neural networks with the ability to discover word-like units that are latent in unsegmented character sequences. In contrast to previous segmentation models that treat word segmentation as an isolated task, our model unifies word discovery, learning how words fit together to form sentences, and, by conditioning the model on visual context, how words' meanings ground in representations of non-linguistic modalities. Experiments show that the unconditional model learns predictive distributions better than character LSTM models, discovers words competitively with nonparametric Bayesian word segmentation models, and that modeling language conditional on visual context improves performance on both.

1 Introduction

How infants discover words that make up their first language is a long-standing question in developmental psychology (Saffran et al., 1996). Machine learning has contributed much to this discussion by showing that predictive models of language are capable of inferring the existence of word boundaries solely based on statistical properties of the input (Elman, 1990; Brent and Cartwright, 1996; Goldwater et al., 2009). However, there are two serious limitations of current models of word learning in the context of the broader problem of language acquisition. First, language acquisition involves not only learning what words there are ("the lexicon"), but also how they fit together ("the grammar"). Unfortunately, the best language models, measured in terms of their ability to predict language (i.e., those which seem acquire grammar best), segment quite poorly (Chung et al., 2017; Wang et al., 2017; Kádár et al., 2018), while the strongest models in terms of word segmentation (Goldwater et al., 2009; Berg-Kirkpatrick et al., 2010) do not adequately account for the long-range dependencies that are manifest in language and that are easily captured by recurrent neural networks (Mikolov et al., 2010). Second, word learning involves not only discovering what words exist and how they fit together grammatically, but also determining their non-linguistic referents, that is, their grounding. The work that has looked at modeling acquisition of grounded language from character sequences—usually in the context of linking words to a visually experienced environment—has either explicitly avoided modeling word units (Gelderloos and Chrupała, 2016) or relied on high-level representations of visual context that overly simplify the richness and ambiguity of the visual signal (Johnson et al., 2010; Räsänen and Rasilo, 2015).

In this paper, we introduce a single model that discovers words, learns how they fit together (not just locally, but across a complete sentence), and grounds them in learned representations of naturalistic non-linguistic visual contexts. We argue that such a unified model is preferable to a pipeline model of language acquisition (e.g., a model where words are learned by one character-aware model, and then a full-sentence grammar is acquired by a second language model using the words predicted by the first). Our preference for the unified model may be expressed in terms of basic notions of simplicity (we require one model rather than two), and in terms of the Continuity Hypothesis of Pinker (1984), which argues that we should assume, absent strong evidence to the contrary, that children have the same cognitive systems as adults, and differences are due to them having set their parameters differently/immaturely.

In §2 we introduce a neural model of sentences that explicitly discovers and models word-like units from completely unsegmented sequences of characters. Since it is a model of complete sentences

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6429–6441
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

(rather than just a word discovery model), and it can incorporate multimodal conditioning context (rather than just modeling language unconditionally), it avoids the two continuity problems identified above. Our model operates by generating text as a sequence of segments, where each segment is generated either character-by-character from a sequence model or as a single draw from a lexical memory of multi-character units. The segmentation decisions and decisions about how to generate words are not observed in the training data and marginalized during learning using a dynamic programming algorithm (§3).

Our model depends crucially on two components. The first is, as mentioned, a lexical memory. This lexicon stores pairs of a vector (key) and a string (value) the strings in the lexicon are contiguous sequences of characters encountered in the training data; and the vectors are randomly initialized and learned during training. The second component is a regularizer (§4) that prevents the model from overfitting to the training data by overusing the lexicon to account for the training data.[1]

Our evaluation (§5–§7) looks at both language modeling performance and the quality of the induced segmentations, in both unconditional (sequence-only) contexts and when conditioning on a related image. First, we look at the segmentations induced by our model. We find that these correspond closely to human intuitions about word segments, competitive with the best existing models for unsupervised word discovery. Importantly, these segments are obtained in models whose hyperparameters are tuned to optimize validation (held-out) likelihood, whereas tuning the hyperparameters of our benchmark models using held-out likelihood produces poor segmentations. Second, we confirm findings (Kawakami et al., 2017; Mielke and Eisner, 2018) that show that word segmentation information leads to better language models compared to pure character models. However, in contrast to previous work, we realize this performance improvement without having to observe the segment boundaries. Thus, our model may be applied straightforwardly to Chinese, where word boundaries are not part of the orthography.

Ablation studies demonstrate that both the lexicon and the regularizer are crucial for good performance, particularly in word segmentation—removing either or both significantly harms performance. In a final experiment, we learn to model language that describes images, and we find that conditioning on visual context improves segmentation performance in our model (compared to the performance when the model does not have access to the image). On the other hand, in a baseline model that predicts boundaries based on entropy spikes in a character-LSTM, making the image available to the model has no impact on the quality of the induced segments, demonstrating again the value of explicitly including a word lexicon in the language model.

2 Model

We now describe the segmental neural language model (SNLM). Refer to Figure 1 for an illustration. The SNLM generates a character sequence $\boldsymbol{x} = x_1, \ldots, x_n$, where each x_i is a character in a finite character set Σ. Each sequence $\boldsymbol{x}$ is the concatenation of a sequence of segments $\underline{\boldsymbol{s}} = \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{|\underline{\boldsymbol{s}}|}$ where $|\underline{\boldsymbol{s}}| \leq n$ measures the length of the sequence in segments and each segment $\boldsymbol{s}_i \in \Sigma^+$ is a sequence of characters, $s_{i,1}, \ldots, s_{i,|\boldsymbol{s}_i|}$. Intuitively, each $\boldsymbol{s}_i$ corresponds to one word. Let $\pi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i)$ represent the concatenation of the characters of the segments $\boldsymbol{s}_1$ to $\boldsymbol{s}_i$, discarding segmentation information; thus $\boldsymbol{x} = \pi(\underline{\boldsymbol{s}})$. For example if $\boldsymbol{x} = \texttt{anapple}$, the underlying segmentation might be $\underline{\boldsymbol{s}} = \texttt{an apple}$ (with $\boldsymbol{s}_1 = \texttt{an}$ and $\boldsymbol{s}_2 = \texttt{apple}$), or $\underline{\boldsymbol{s}} = \texttt{a nap ple}$, or any of the $2^{|\boldsymbol{x}|-1}$ segmentation possibilities for $\boldsymbol{x}$.

The SNLM defines the distribution over $\boldsymbol{x}$ as the marginal distribution over all segmentations that give rise to $\boldsymbol{x}$, i.e.,

$$p(\boldsymbol{x}) = \sum_{\underline{\boldsymbol{s}}:\pi(\underline{\boldsymbol{s}})=\boldsymbol{x}} p(\underline{\boldsymbol{s}}). \tag{1}$$

To define the probability of $p(\underline{\boldsymbol{s}})$, we use the chain rule, rewriting this in terms of a product of the series of conditional probabilities, $p(\boldsymbol{s}_t \mid \underline{\boldsymbol{s}}_{<t})$. The process stops when a special end-sequence segment $\langle/\mathrm{S}\rangle$ is generated. To ensure that the summation in Eq. 1 is tractable, we assume the following:

$$p(\boldsymbol{s}_t \mid \underline{\boldsymbol{s}}_{<t}) \approx p(\boldsymbol{s}_t \mid \pi(\underline{\boldsymbol{s}}_{<t})) = p(\boldsymbol{s}_t \mid \boldsymbol{x}_{<t}), \tag{2}$$

which amounts to a conditional semi-Markov assumption—i.e., non-Markovian generation hap-

[1] Since the lexical memory stores strings that appear in the training data, each sentence could, in principle, be generated as a single lexical unit, thus the model could fit the training data perfectly while generalizing poorly. The regularizer penalizes based on the expectation of the powered length of each segment, preventing this degenerate solution from being optimal.

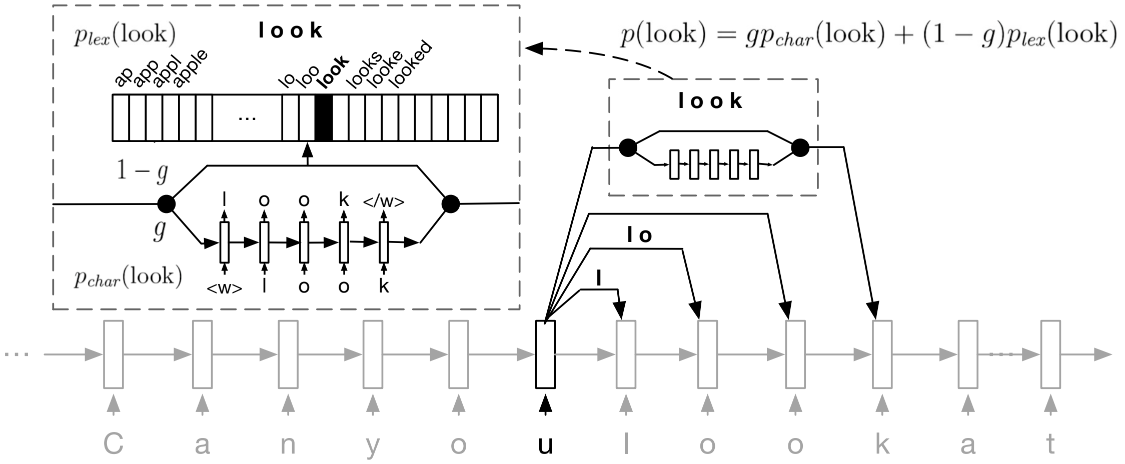

Figure 1: Fragment of the segmental neural language model while evaluating the marginal likelihood of a sequence. At the indicated time, the model has generated the sequence *Canyou*, and four possible continuations are shown.

pens inside each segment, but the segment generation probability does not depend on memory of the previous segmentation decisions, only upon the sequence of characters $\pi(\boldsymbol{s}_{<t})$ corresponding to the prefix character sequence $\boldsymbol{x}_{<t}$. This assumption has been employed in a number of related models to permit the use of LSTMs to represent rich history while retaining the convenience of dynamic programming inference algorithms (Wang et al., 2017; Ling et al., 2017; Graves, 2012).

2.1 Segment generation

We model $p(\boldsymbol{s}_t \mid \boldsymbol{x}_{<t})$ as a mixture of two models, one that generates the segment using a sequence model and the other that generates multi-character sequences as a single event. Both are conditional on a common representation of the history, as is the mixture proportion.

Representing history To represent $\boldsymbol{x}_{<t}$, we use an LSTM encoder to read the sequence of characters, where each character type $\sigma \in \Sigma$ has a learned vector embedding $\mathbf{v}_\sigma$. Thus the history representation at time t is $\mathbf{h}_t = \text{LSTM}_{enc}(\mathbf{v}_{x_1}, \ldots, \mathbf{v}_{x_t})$. This corresponds to the standard history representation for a character-level language model, although in general, we assume that our modelled data is not delimited by whitespace.

Character-by-character generation The first component model, $p_{char}(\boldsymbol{s}_t \mid \mathbf{h}_t)$, generates $\boldsymbol{s}_t$ by sampling a sequence of characters from a LSTM language model over Σ and a two extra special symbols, an end-of-word symbol $\langle/\text{w}\rangle \notin \Sigma$ and the end-of-sequence symbol $\langle/\text{s}\rangle$ discussed above. The initial state of the LSTM is a learned transformation of $\mathbf{h}_t$, the initial cell is $\mathbf{0}$, and different parameters than the history encoding LSTM are used. During generation, each letter that is sampled (i.e., each $s_{t,i}$) is fed back into the LSTM in the usual way and the probability of the character sequence decomposes according to the chain rule. The end-of-sequence symbol can never be generated in the initial position.

Lexical generation The second component model, $p_{lex}(\boldsymbol{s}_t \mid \mathbf{h}_t)$, samples full segments from lexical memory. Lexical memory is a key-value memory containing M entries, where each key, $\mathbf{k}_i$, a vector, is associated with a value $\boldsymbol{v}_i \in \Sigma^+$. The generation probability of $\boldsymbol{s}_t$ is defined as

$$\begin{aligned}\mathbf{h}'_t &= \text{MLP}(\mathbf{h}_t)\\ \mathbf{m} &= \text{softmax}(\mathbf{K}\mathbf{h}'_t + \mathbf{b})\\ p_{lex}(\boldsymbol{s}_t \mid \mathbf{h}_t) &= \sum_{i=1}^{M} m_i[\boldsymbol{v}_i = \boldsymbol{s}_t],\end{aligned}$$

where $[\boldsymbol{v}_i = \boldsymbol{s}_t]$ is 1 if the ith value in memory is $\boldsymbol{s}_t$ and 0 otherwise, and $\mathbf{K}$ is a matrix obtained by stacking the $\mathbf{k}_i^\top$'s. This generation process assigns zero probability to most strings, but the alternate character model can generate all of Σ^+.

In this work, we fix the $\boldsymbol{v}_i$'s to be subsequences of at least length 2, and up to a maximum length L that are observed at least F times in the training data. These values are tuned as hyperparameters (See Appendix C for details of the experiments).

Mixture proportion The mixture proportion, g_t, determines how likely the character generator is to be used at time t (the lexicon is used with probability $1 - g_t$). It is defined by as $g_t = \sigma(\mathrm{MLP}(\mathbf{h}_t))$.

Total segment probability The total generation probability of $\boldsymbol{s}_t$ is thus

$$p(\boldsymbol{s}_t \mid \boldsymbol{x}_{<t}) = g_t p_{char}(\boldsymbol{s}_t \mid \mathbf{h}_t) + (1 - g_t) p_{lex}(\boldsymbol{s}_t \mid \mathbf{h}_t).$$

3 Inference

We are interested in two inference questions: first, given a sequence $\boldsymbol{x}$, evaluate its (log) marginal likelihood; second, given $\boldsymbol{x}$, find the most likely decomposition into segments $\underline{\boldsymbol{s}}^*$.

Marginal likelihood To efficiently compute the marginal likelihood, we use a variant of the forward algorithm for semi-Markov models (Yu, 2010), which incrementally computes a sequence of probabilities, α_i, where α_i is the marginal likelihood of generating $\boldsymbol{x}_{\leq i}$ and concluding a segment at time i. Although there are an exponential number of segmentations of $\boldsymbol{x}$, these values can be computed using $O(|\boldsymbol{x}|)$ space and $O(|\boldsymbol{x}|^2)$ time as:

$$\alpha_0 = 1, \qquad \alpha_t = \sum_{j=t-L}^{t-1} \alpha_j p(\boldsymbol{s} = \boldsymbol{x}_{j:t} \mid \boldsymbol{x}_{<j}). \tag{3}$$

By letting $x_{t+1} = \langle/\mathrm{S}\rangle$, then $p(\boldsymbol{x}) = \alpha_{t+1}$.

Most probable segmentation The most probable segmentation of a sequence $\boldsymbol{x}$ can be computed by replacing the summation with a $\max$ operator in Eq. 3 and maintaining backpointers.

4 Expected length regularization

When the lexical memory contains all the substrings in the training data, the model easily overfits by copying the longest continuation from the memory. To prevent overfitting, we introduce a regularizer that penalizes based on the expectation of the exponentiated (by a hyperparameter β) length of each segment:

$$R(\boldsymbol{x}, \beta) = \sum_{\underline{\boldsymbol{s}}:\pi(\underline{\boldsymbol{s}})=\boldsymbol{x}} p(\underline{\boldsymbol{s}} \mid \boldsymbol{x}) \sum_{\boldsymbol{s} \in \underline{\boldsymbol{s}}} |\boldsymbol{s}|^{\beta}.$$

This can be understood as a regularizer based on the double exponential prior identified to be effective in previous work (Liang and Klein, 2009; Berg-Kirkpatrick et al., 2010). This expectation is a differentiable function of the model parameters. Because of the linearity of the penalty across segments, it can be computed efficiently using the above dynamic programming algorithm under the expectation semiring (Eisner, 2002). This is particularly efficient since the expectation semiring jointly computes the expectation and marginal likelihood in a single forward pass. For more details about computing gradients of expectations under distributions over structured objects with dynamic programs and semirings, see Li and Eisner (2009).

4.1 Training Objective

The model parameters are trained by minimizing the penalized log likelihood of a training corpus $\mathcal{D}$ of unsegmented sentences,

$$\mathcal{L} = \sum_{\boldsymbol{x} \in \mathcal{D}} [-\log p(\boldsymbol{x}) + \lambda R(\boldsymbol{x}, \beta)].$$

5 Datasets

We evaluate our model on both English and Chinese segmentation. For both languages, we used standard datasets for word segmentation and language modeling. We also use MS-COCO to evaluate how the model can leverage conditioning context information. For all datasets, we used train, validation and test splits.[2] Since our model assumes a closed character set, we removed validation and test samples which contain characters that do not appear in the training set. In the English corpora, whitespace characters are removed. In Chinese, they are not present to begin with. Refer to Appendix A for dataset statistics.

5.1 English

Brent Corpus The Brent corpus is a standard corpus used in statistical modeling of child language acquisition (Brent, 1999; Venkataraman, 2001).[3] The corpus contains transcriptions of utterances directed at 13- to 23-month-old children. The corpus has two variants: an orthographic one (**BR-text**) and a phonemic one (**BR-phono**), where each character corresponds to a single English phoneme. As the Brent corpus does not have a standard train and test split, and we want to tune the parameters by measuring the fit to held-out data, we used the first 80% of the utterances for training and the next 10% for validation and the rest for test.

[2] The data and splits used are available at `https://s3.eu-west-2.amazonaws.com/k-kawakami/seg.zip`.

[3] `https://childes.talkbank.org/derived`

English Penn Treebank (PTB) We use the commonly used version of the PTB prepared by Mikolov et al. (2010). However, since we removed space symbols from the corpus, our cross entropy results cannot be compared to those usually reported on this dataset.

5.2 Chinese

Since Chinese orthography does not mark spaces between words, there have been a number of efforts to annotate word boundaries. We evaluate against two corpora that have been manually segmented according different segmentation standards.

Beijing University Corpus (PKU) The Beijing University Corpus was one of the corpora used for the International Chinese Word Segmentation Bakeoff (Emerson, 2005).

Chinese Penn Treebank (CTB) We use the Penn Chinese Treebank Version 5.1 (Xue et al., 2005). It generally has a coarser segmentation than PKU (e.g., in CTB a full name, consisting of a given name and family name, is a single token), and it is a larger corpus.

5.3 Image Caption Dataset

To assess whether jointly learning about meanings of words from non-linguistic context affects segmentation performance, we use image and caption pairs from the COCO caption dataset (Lin et al., 2014). We use 10,000 examples for both training and testing and we only use one reference per image. The images are used to be conditional context to predict captions. Refer to Appendix B for the dataset construction process.

6 Experiments

We compare our model to benchmark Bayesian models, which are currently the best known unsupervised word discovery models, as well as to a simple deterministic segmentation criterion based on surprisal peaks (Elman, 1990) on language modeling and segmentation performance. Although the Bayeisan models are shown to able to discover plausible word-like units, we found that a set of hyperparameters that provides best performance with such model on language modeling does not produce good structures as reported in previous works. This is problematic since there is no objective criteria to find hyperparameters in fully unsupervised manner when the model is applied to completely unknown languages or domains. Thus, our experiments are designed to assess how well the models infers word segmentations of unsegmented inputs when they are trained and tuned to maximize the likelihood of the held-out text.

DP/HDP Benchmarks Among the most effective existing word segmentation models are those based on hierarchical Dirichlet process (HDP) models (Goldwater et al., 2009; Teh et al., 2006) and hierarchical Pitman–Yor processes (Mochihashi et al., 2009). As a representative of these, we use a simple bigram HDP model:

$$\begin{aligned}
\theta_{\cdot} &\sim \mathrm{DP}(\alpha_0, p_0) \\
\theta_{\cdot|\boldsymbol{s}} &\sim \mathrm{DP}(\alpha_1, \theta_{\cdot}) \qquad \forall \boldsymbol{s} \in \Sigma^* \\
\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t &\sim \mathrm{Categorical}(\theta_{\cdot|\boldsymbol{s}_t}).
\end{aligned}$$

The base distribution, p_0, is defined over strings in $\Sigma^* \cup \{\langle/\mathrm{S}\rangle\}$ by deciding with a specified probability to end the utterance, a geometric length model, and a uniform probability over Σ at a each position. Intuitively, it captures the preference for having short words in the lexicon. In addition to the HDP model, we also evaluate a simpler single Dirichlet process (DP) version of the model, in which the $\boldsymbol{s}_t$'s are generated directly as draws from $\mathrm{Categorical}(\theta_{\cdot})$. We use an empirical Bayesian approach to select hyperparameters based on the likelihood assigned by the inferred posterior to a held-out validation set. Refer to Appendix D for details on inference.

Deterministic Baselines Incremental word segmentation is inherently ambiguous (e.g., the letters *the* might be a single word, or they might be the beginning of the longer word *theater*). Nevertheless, several deterministic functions of prefixes have been proposed in the literature as strategies for discovering rudimentary word-like units hypothesized for being useful for bootstrapping the lexical acquisition process or for improving a model's predictive accuracy. These range from surprisal criteria (Elman, 1990) to sophisticated language models that switch between models that capture intra- and inter-word dynamics based on deterministic functions of prefixes of characters (Chung et al., 2017; Shen et al., 2018).

In our experiments, we also include such deterministic segmentation results using (1) the surprisal criterion of Elman (1990) and (2) a two-level hierarchical multiscale LSTM (Chung et al., 2017), which has been shown to predict boundaries in whitespace-containing character sequences at positions corresponding to word boundaries. As with all experiments in this paper, the BR-corpora for this experiment do not contain spaces.

SNLM Model configurations and Evaluation LSTMs had 512 hidden units with parameters learned using the Adam update rule (Kingma and Ba, 2015). We evaluated our models with bits-per-character (bpc) and segmentation accuracy (Brent, 1999; Venkataraman, 2001; Goldwater et al., 2009). Refer to Appendices C–F for details of model configurations and evaluation metrics.

For the image caption dataset, we extend the model with a standard attention mechanism in the backbone LSTM (LSTM$_{enc}$) to incorporate image context. For every character-input, the model calculates attentions over image features and use them to predict the next characters. As for image representations, we use features from the last convolution layer of a pre-trained VGG19 model (Simonyan and Zisserman, 2014).

7 Results

In this section, we first do a careful comparison of segmentation performance on the phonemic Brent corpus (BR-phono) across several different segmentation baselines, and we find that our model obtains competitive segmentation performance. Additionally, ablation experiments demonstrate that both lexical memory and the proposed expected length regularization are necessary for inferring good segmentations. We then show that also on other corpora, we likewise obtain segmentations better than baseline models. Finally, we also show that our model has superior performance, in terms of held-out perplexity, compared to a character-level LSTM language model. Thus, overall, our results show that we can obtain good segmentations on a variety of tasks, while still having very good language modeling performance.

Word Segmentation (BR-phono) Table 1 summarizes the segmentation results on the widely used BR-phono corpus, comparing it to a variety of baselines. **Unigram DP**, **Bigram HDP**, **LSTM suprisal** and **HMLSTM** refer to the benchmark models explained in §6. The ablated versions of our model show that without the lexicon (−memory), without the expected length penalty (−length), and without either, our model fails to discover good segmentations. Furthermore, we draw attention to the difference in the performance of the HDP and DP models when using subjective settings of the hyperparameters and the empirical settings (likelihood). Finally, the deterministic baselines are interesting in two ways. First, LSTM surprisal is a remarkably good heuristic for segmenting text (although we will see below that its performance is much less good on other datasets). Second, despite careful tuning, the HMLSTM of Chung et al. (2017) fails to discover good segments, although in their paper they show that when spaces are present between, HMLSTMs learn to switch between their internal models in response to them.

Furthermore, the priors used in the DP/HDP models were tuned to maximize the likelihood assigned to the validation set by the inferred posterior predictive distribution, in contrast to previous papers which either set them subjectively or inferred them (Johnson and Goldwater, 2009). For example, the DP and HDP model with subjective priors obtained 53.8 and 72.3 F1 scores, respectively (Goldwater et al., 2009). However, when the hyperparameters are set to maximize held-out likelihood, this drops obtained 56.1 and 56.9. Another result on this dataset is the feature unigram model of Berg-Kirkpatrick et al. (2010), which obtains an 88.0 F1 score with hand-crafted features and by selecting the regularization strength to optimize segmentation performance. Once the features are removed, the model achieved a 71.5 F1 score when it is tuned on segmentation performance and only 11.5 when it is tuned on held-out likelihood.

	P	R	F1
LSTM surprisal (Elman, 1990)	54.5	55.5	55.0
HMLSTM (Chung et al., 2017)	8.1	13.3	10.1
Unigram DP	63.3	50.4	56.1
Bigram HDP	53.0	61.4	56.9
SNLM (−memory, −length)	54.3	34.9	42.5
SNLM (+memory, −length)	52.4	36.8	43.3
SNLM (−memory, +length)	57.6	43.4	49.5
SNLM (+memory, +length)	**81.3**	**77.5**	**79.3**

Table 1: Summary of segmentation performance on phoneme version of the Brent Corpus (**BR-phono**).

Word Segmentation (other corpora) Table 2 summarizes results on the BR-text (orthographic Brent corpus) and Chinese corpora. As in the previous section, all the models were trained to maximize held-out likelihood. Here we observe a similar pattern, with the SNLM outperforming the baseline models, despite the tasks being quite different from each other and from the BR-phono task.

		P	R	F1
BR-text	LSTM surprisal	36.4	49.0	41.7
	Unigram DP	64.9	55.7	60.0
	Bigram HDP	52.5	63.1	57.3
	SNLM	**68.7**	**78.9**	**73.5**
PTB	LSTM surprisal	27.3	36.5	31.2
	Unigram DP	51.0	49.1	50.0
	Bigram HDP	34.8	47.3	40.1
	SNLM	**54.1**	**60.1**	**56.9**
CTB	LSTM surprisal	41.6	25.6	31.7
	Unigram DP	61.8	49.6	55.0
	Bigram HDP	67.3	67.7	67.5
	SNLM	**78.1**	**81.5**	**79.8**
PKU	LSTM surprisal	38.1	23.0	28.7
	Unigram DP	60.2	48.2	53.6
	Bigram HDP	66.8	67.1	66.9
	SNLM	**75.0**	**71.2**	**73.1**

Table 2: Summary of segmentation performance on other corpora.

Word Segmentation Qualitative Analysis We show some representative examples of segmentations inferred by various models on the BR-text and PKU corpora in Table 3. As reported in Goldwater et al. (2009), we observe that the DP models tend to undersegment, keep long frequent sequences together (e.g., they failed to separate articles). HDPs do successfully prevent oversegmentation; however, we find that when trained to optimize held-out likelihood, they often insert unnecessary boundaries between words, such as *yo u*. Our model's performance is better, but it likewise shows a tendency to oversegment. Interestingly, we can observe a tendency tends to put boundaries between morphemes in morphologically complex lexical items such as *dumpty 's*, and *go ing*. Since morphemes are the minimal units that carry meaning in language, this segmentation, while incorrect, is at least plasuible. Turning to the Chinese examples, we see that both baseline models fail to discover basic words such as 山间 (mountain) and 人们 (human).

Finally, we observe that none of the models successfully segment dates or numbers containing multiple digits (all oversegment). Since number types tend to be rare, they are usually not in the lexicon, meaning our model (and the H/DP baselines) must generate them as character sequences.

Language Modeling Performance The above results show that the SNLM infers good word segmentations. We now turn to the question of how well it predicts held-out data. Table 4 summarizes the results of the language modeling experiments. Again, we see that SNLM outperforms the Bayesian models and a character LSTM. Although there are numerous extensions to LSTMs to improve language modeling performance, LSTMs remain a strong baseline (Melis et al., 2018).

One might object that because of the lexicon, the SNLM has many more parameters than the character-level LSTM baseline model. However, unlike parameters in LSTM recurrence which are used every timestep, our memory parameters are accessed very sparsely. Furthermore, we observed that an LSTM with twice the hidden units did not improve the baseline with 512 hidden units on both phonemic and orthographic versions of Brent corpus but the lexicon could. This result suggests more hidden units are useful if the model does not have enough capacity to fit larger datasets, but that the memory structure adds other dynamics which are not captured by large recurrent networks.

Multimodal Word Segmentation Finally, we discuss results on word discovery with non-linguistic context (image). Although there is much evidence that neural networks can reliably learn to exploit additional relevant context to improve language modeling performance (e.g. machine translation and image captioning), it is still unclear whether the conditioning context help to discover *structure* in the data. We turn to this question here. Table 5 summarizes language modeling and segmentation performance of our model and a baseline character-LSTM language model on the COCO image caption dataset. We use the Elman Entropy criterion to infer the segmentation points from the baseline LM, and the MAP segmentation under our model. Again, we find our model outperforms the baseline model in terms of both language modeling and word segmentation accuracy. Interestingly, we find while conditioning on image context leads to reductions in perplexity in both models, in our model the presence of the image further improves segmentation accuracy. This suggests that

		Examples
BR-text	Reference	are you going to make him pretty this morning
	Unigram DP	areyou goingto makehim pretty this morning
	Bigram HDP	areyou go ingto make him p retty this mo rn ing
	SNLM	are you go ing to make him pretty this morning
	Reference	would you like to do humpty dumpty’s button
	Unigram DP	wouldyoul iketo do humpty dumpty ’s button
	Bigram HDP	would youlike to do humptyd umpty ’s butt on
	SNLM	would you like to do humpty dumpty ’s button
PKU	Reference	笑声 、 掌声 、 欢呼声 ， 在 山间 回荡 ， 勾 起 了 人们 对 往事 的 回忆 。
	Unigram DP	笑声 、 掌声 、 欢呼 声 ，在 山 间 回荡 ， 勾 起了 人们对 往事 的 回忆 。
	Bigram HDP	笑 声、 掌声 、 欢 呼声 ，在 山 间 回 荡， 勾 起了 人 们对 往事 的 回忆 。
	SNLM	笑声、 掌声 、 欢呼声 ， 在 山间 回荡 ， 勾起 了 人们 对 往事 的 回忆 。
	Reference	不得 在 江河 电缆 保护区 内 抛锚 、 拖锚 、 炸鱼 、 挖沙 。
	Unigram DP	不得 在 江河电缆 保护 区内抛锚、 拖锚 、炸鱼、挖沙 。
	Bigram HDP	不得 在 江 河 电缆 保护 区内 抛 锚、拖 锚 、 炸鱼、 挖沙 。
	SNLM	不得 在 江河 电缆 保护区 内 抛锚 、 拖锚、 炸鱼 、 挖沙 。

Table 3: Examples of predicted segmentations on English and Chinese.

	BR-text	BR-phono	PTB	CTB	PKU
Unigram DP	2.33	2.93	2.25	6.16	6.88
Bigram HDP	1.96	2.55	1.80	5.40	6.42
LSTM	2.03	2.62	1.65	4.94	6.20
SNLM	**1.94**	**2.54**	**1.56**	**4.84**	**5.89**

Table 4: Test language modeling performance (bpc).

	bpc↓	P↑	R↑	F1↑
Unigram DP	2.23	44.0	40.0	41.9
Bigram HDP	1.68	30.9	40.8	35.1
LSTM (−image)	1.55	31.3	38.2	34.4
SNLM (−image)	1.52	39.8	55.3	46.3
LSTM (+image)	1.42	31.7	39.1	35.0
SNLM (+image)	**1.38**	**46.4**	**62.0**	**53.1**

Table 5: Language modeling (bpc) and segmentation accuracy on COCO dataset. +image indicates that the model has access to image context.

our model and its learning mechanism interact with the conditional context differently than the LSTM does.

To understand what kind of improvements in segmentation performance the image context leads to, we annotated the tokens in the references with part-of-speech (POS) tags and compared relative improvements on recall between SNLM (−image) and SNLM (+image) among the five POS tags which appear more than 10,000 times. We observed improvements on ADJ (+4.5%), NOUN (+4.1%), VERB (+3.1%). The improvements on the categories ADP (+0.5%) and DET (+0.3%) are were more limited. The categories where we see the largest improvement in recall correspond to those that are likely *a priori* to correlate most reliably with observable features. Thus, this result is consistent with a hypothesis that the lexican is successfully acquiring knowledge about how words idiosyncratically link to visual features.

Segmentation State-of-the-Art The results reported are not the best-reported numbers on the English phoneme or Chinese segmentation tasks. As we discussed in the introduction, previous work has focused on segmentation in isolation from language modeling performance. Models that obtain better segmentations include the adaptor grammars (F1: 87.0) of Johnson and Goldwater (2009) and the feature-unigram model (88.0) of Berg-Kirkpatrick et al. (2010). While these results are better in terms of segmentation, they are weak language models (the feature unigram model is effectively a unigram word model; the adaptor grammar model is effectively phrasal unigram model; both are incapable of generalizing about substantially non-local dependencies). Additionally, the features and grammars used in prior work reflect certain English-specific design considerations (e.g., syllable structure in the case of adaptor grammars and phonotactic equivalence classes in the feature unigram model), which make them questionable models if the goal is to explore what models and biases enable word discovery in general. For Chinese, the best nonparametric models perform better at segmentation (Zhao and Kit, 2008; Mochihashi et al., 2009), but again they are weaker language models than neural models. The neural model of Sun and Deng (2018) is similar to our model without lexical memory or length regularization; it obtains 80.2 F1 on the PKU dataset; however, it uses gold segmentation data during training and hyperparameter selection,[4] whereas our approach requires no gold standard segmentation data.

8 Related Work

Learning to discover and represent temporally extended structures in a sequence is a fundamental problem in many fields. For example in language processing, unsupervised learning of multiple levels of linguistic structures such as morphemes (Snyder and Barzilay, 2008), words (Goldwater et al., 2009; Mochihashi et al., 2009; Wang et al., 2014) and phrases (Klein and Manning, 2001) have been investigated. Recently, speech recognition has benefited from techniques that enable the discovery of subword units (Chan et al., 2017; Wang et al., 2017); however, in that work, the optimally discovered character sequences look quite unlike orthographic words. In fact, the model proposed by Wang et al. (2017) is essentially our model without a lexicon or the expected length regularization, i.e., (−memory, −length), which we have shown performs quite poorly in terms of segmentation accuracy. Finally, some prior work has also sought to discover lexical units directly from speech based on speech-internal statistical regularities (Kamper et al., 2016), as well as jointly with grounding (Chrupała et al., 2017).

9 Conclusion

Word discovery is a fundamental problem in language acquisition. While work studying the problem in isolation has provided valuable insights (showing both what data is sufficient for word discovery with which models), this paper shows that neural models offer the flexibility and performance to productively study the various facets of the problem in a more unified model. While this work unifies several components that had previously been studied in isolation, our model assumes access to phonetic categories. The development of these categories likely interact with the development of the lexicon and acquisition of semantics (Feldman et al., 2013; Fourtassi and Dupoux, 2014), and thus subsequent work should seek to unify more aspects of the acquisition problem.

Acknowledgments

We thank Mark Johnson, Sharon Goldwater, and Emmanuel Dupoux, as well as many colleagues at DeepMind, for their insightful comments and suggestions for improving this work and the resulting paper.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proc. NAACL*.

Michael R Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1):71–105.

Michael R Brent and Timothy A Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1):93–125.

William Chan, Yu Zhang, Quoc Le, and Navdeep Jaitly. 2017. Latent sequence decompositions. In *Proc. ICLR*.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. 2017. Representations of language in a model of visually grounded speech signal.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *Proc. ICLR*.

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proc. ACL*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proc. SIGHAN Workshop*.

Naomi H. Feldman, Thomas L. Griffiths, Sharon Goldwater, and James L. Morgan. 2013. A role for the developing lexicon in phonetic category acquisition. *Psychological Review*, 120(4):751–778.

Abdellah Fourtassi and Emmanuel Dupoux. 2014. A rudimentary lexicon and semantics help bootstrap phoneme acquisition. In *Proc. EMNLP*.

[4] `https://github.com/Edward-Sun/SLM/blob/d37ad735a7b1d5af430b96677c2ecf37a65f59b7/codes/run.py#L329`

Lieke Gelderloos and Grzegorz Chrupała. 2016. From phonemes to images: levels of representation in a recurrent neural model of visually-grounded language learning. In *Proc. COLING*.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Mark Johnson, Katherine Demuth, Michael Frank, and Bevan K. Jones. 2010. Synergies in learning words and their referents. In *Proc. NIPS*.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proc. NAACL*, pages 317–325.

Ákos Kádár, Marc-Alexandre Côté, Grzegorz Chrupała, and Afra Alishahi. 2018. Revisiting the hierarchical multiscale LSTM. In *Proc. COLING*.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016. Unsupervised word segmentation and lexicon induction discovery using acoustic word embeddings. *IEEE Transactions on Audio, Speech, and Language Processing*, 24(4):669–679.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2017. Learning to create and reuse words in open-vocabulary neural language modeling. In *Proc. ACL*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Dan Klein and Christopher D Manning. 2001. Distributional phrase structure induction. In *Workshop Proc. ACL*.

Zhifei Li and Jason Eisner. 2009. First-and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proc. EMNLP*.

Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *Proc. NAACL*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proc. ECCV*, pages 740–755.

Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Andrew Senior, Fumin Wang, and Phil Blunsom. 2017. Latent predictor networks for code generation. In *Proc. ACL*.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *Proc. ICLR*.

Sebastian J. Mielke and Jason Eisner. 2018. Spell once, summon anywhere: A two-level open-vocabulary language model. In *Proc. NAACL*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proc. Interspeech*.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman–Yor language modeling.

Steven Pinker. 1984. *Language learnability and language development*. Harvard University Press.

Okko Räsänen and Heikki Rasilo. 2015. A joint model of word segmentation and meaning acquisition through cross-situational learning. *Psychological Review*, 122(4):792–829.

Jenny R Saffran, Richard N Aslin, and Elissa L Newport. 1996. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *Proc. ICLR*.

K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proc. ACL*.

Zhiqing Sun and Zhi-Hong Deng. 2018. Unsupervised neural word segmentation for Chinese via segmental language modeling.

Yee-Whye Teh, Michael I. Jordan, Matthew J. Beal, and Daivd M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Anand Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):351–372.

Chong Wang, Yining Wan, Po-Sen Huang, Abdelrahman Mohammad, Dengyong Zhou, and Li Deng. 2017. Sequence modeling via segmentations. In *Proc. ICML*.

Xiaolin Wang, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2014. Empirical study of unsupervised Chinese word segmentation methods for SMT on large-scale corpora.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.

Shun-Zheng Yu. 2010. Hidden semi-Markov models. *Artificial Intelligence*, 174(2):215–243.

Hai Zhao and Chunyu Kit. 2008. An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework. In *Proc. IJCNLP*.

A Dataset statistics

Table 6 summarizes dataset statistics.

B Image Caption Dataset Construction

We use 8000, 2000 and 10000 images for train, development and test set in order of integer ids specifying image in cocoapi[5] and use first annotation provided for each image. We will make pairs of image id and annotation id available from `https://s3.eu-west-2.amazonaws.com/k-kawakami/seg.zip`.

C SNLM Model Configuration

For each RNN based model we used 512 dimensions for the character embeddings and the LSTMs have 512 hidden units. All the parameters, including character projection parameters, are randomly sampled from uniform distribution from -0.08 to 0.08. The initial hidden and memory state of the LSTMs are initialized with zero. A dropout rate of 0.5 was used for all but the recurrent connections.

To restrict the size of memory, we stored substrings which appeared F-times in the training corpora and tuned F with grid search. The maximum length of subsequences L was tuned on the held-out likelihood using a grid search. Tab. 7 summarizes the parameters for each dataset. Note that we did not tune the hyperparameters on segmentation quality to ensure that the models are trained in a purely unsupervised manner assuming no reference segmentations are available.

D DP/HDP Inference

By integrating out the draws from the DP's, it is possible to do inference using Gibbs sampling directly in the space of segmentation decisions. We use 1,000 iterations with annealing to find an approximation of the MAP segmentation and then use the corresponding posterior predictive distribution to estimate the held-out likelihood assigned by the model, marginalizing the segmentations using appropriate dynamic programs. The evaluated segmentation was the most probable segmentation according to the posterior predictive distribution.

In the original Bayesian segmentation work, the hyperparameters (i.e., α_0, α_1, and the components of p_0) were selected subjectively. To make comparison with our neural models fairer, we instead used an empirical approach and set them using the held-out likelihood of the validation set. However, since this disadvantages the DP/HDP models in terms of segmentation, we also report the original results on the BR corpora.

E Learning

The models were trained with the Adam update rule (Kingma and Ba, 2015) with a learning rate of 0.01. The learning rate is divided by 4 if there is no improvement on development data. The maximum norm of the gradients was clipped at 1.0.

F Evaluation Metrics

Language Modeling We evaluated our models with bits-per-character (bpc), a standard evaluation metric for character-level language models. Following the definition in Graves (2013), bits-per-character is the average value of $-\log_2 p(x_t \mid \boldsymbol{x}_{<t})$ over the whole test set,

$$bpc = -\frac{1}{|\boldsymbol{x}|} \log_2 p(\boldsymbol{x}),$$

where $|\boldsymbol{x}|$ is the length of the corpus in characters. The bpc is reported on the test set.

Segmentation We also evaluated segmentation quality in terms of precision, recall, and F1 of word tokens (Brent, 1999; Venkataraman, 2001; Goldwater et al., 2009). To get credit for a word, the models must correctly identify both the left and right boundaries. For example, if there is a pair of a reference segmentation and a prediction,

Reference: `do you see a boy`
Prediction: `doyou see a boy`

then 4 words are discovered in the prediction where the reference has 5 words. 3 words in the prediction match with the reference. In this case, we report scores as precision = 75.0 (3/4), recall = 60.0 (3/5), and F1, the harmonic mean of precision and recall, 66.7 (2/3). To facilitate comparison with previous work, segmentation results are reported on the union of the training, validation, and test sets.

[5] https://github.com/cocodataset/cocoapi

	Sentence			Char. Types			Word Types			Characters			Average Word Length		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
BR-text	7832	979	979	30	30	29	1237	473	475	129k	16k	16k	3.82	4.06	3.83
BR-phono	7832	978	978	51	51	50	1183	457	462	104k	13k	13k	2.86	2.97	2.83
PTB	42068	3370	3761	50	50	48	10000	6022	6049	5.1M	400k	450k	4.44	4.37	4.41
CTB	50734	349	345	160	76	76	60095	1769	1810	3.1M	18k	22k	4.84	5.07	5.14
PKU	17149	1841	1790	90	84	87	52539	13103	11665	2.6M	247k	241k	4.93	4.94	4.85
COCO	8000	2000	10000	50	42	48	4390	2260	5072	417k	104k	520k	4.00	3.99	3.99

Table 6: Summary of Dataset Statistics.

	max len (L)	min freq (F)	λ
BR-text	10	10	7.5e-4
BR-phono	10	10	9.5e-4
PTB	10	100	5.0e-5
CTB	5	25	1.0e-2
PKU	5	25	9.0e-3
COCO	10	100	2.0e-4

Table 7: Hyperparameter values used.