Less is Better: A cognitively inspired unsupervised model for language segmentation

Jinbiao Yang
Max Planck Institute for Psycholinguistics
Centre for Language Studies, Radboud University
jinbiao.yang@mpi.nl

Antal van den Bosch
KNAW Meertens Institute
antal.van.den.bosch
@meertens.knaw.nl

Stefan L. Frank
Centre for Language Studies, Radboud University
s.frank@let.ru.nl

Abstract

Language users process utterances by segmenting them into many *cognitive units*, which vary in their sizes and linguistic levels. Although we can do such unitization/segmentation easily, its cognitive mechanism is still not clear. This paper proposes an unsupervised model, *Less-is-Better* (LiB), to simulate the human cognitive process with respect to language unitization/segmentation. LiB follows the principle of least effort and aims to build a lexicon which minimizes the number of unit tokens (alleviating the effort of analysis) and number of unit types (alleviating the effort of storage) at the same time on any given corpus. LiB's workflow is inspired by empirical cognitive phenomena. The design makes the mechanism of LiB cognitively plausible and the computational requirement light-weight. The lexicon generated by LiB performs the best among different types of lexicons (e.g. ground-truth words) both from an information-theoretical view and a cognitive view, which suggests that the LiB lexicon may be a plausible proxy of the mental lexicon.

1 Introduction

During language comprehension, we cannot always process an utterance instantly. Instead, we need to segment all but the shortest pieces of text or speech into smaller chunks. Since these chunks are likely the cognitive processing units for language understanding, we call them *cognitive units* in this paper. A chunk may be any string of letters, characters, or phonemes that occurs in the language, but which chunks serve as the cognitive units? Traditional studies (Chomsky, 1957; Taft, 2013, for example) often use words as the units in sentence analysis. But speech, as well as some writing systems such as Chinese, lack a clear word boundary. Even for written languages which use spaces as word boundaries, psychological evidence indicates that the morphemes, which are sub-word units, in infrequent or opaque compound words take priority over the whole word (Fiorentino et al., 2014; MacGregor and Shtyrov, 2013); at the same time, some supra-word units such as frequent phrases and idioms are also stored in our long-term mental lexicon (Arnon and Snider, 2010; Bannard and Matthews, 2008; Jackendoff, 2002). The evidence suggests that the cognitive units can be of different sizes; they can be words, or smaller than words, or multi-word expressions.

Despite the flexible size of the cognitive units, and the lack of overt segmentation clues, infants are able to implicitly learn the units in their caregivers' speech, and then generate their own utterances. Arguably, children's language intelligence allows them to build their own lexicons from zero knowledge about the basic (cognitive) units in the particular language the child is learning, and then use the lexicon to segment language sequences. Can we mimic this ability of a human language learner in a computer model? This question is often phrased as the task of unsupervised segmentation. Several types of computational models or NLP algorithms have been proposed for segmentation, taking different approaches:

Proceedings of the Workshop on Cognitive Aspects of the Lexicon, pages 33–45
Barcelona, Spain (Online), December 12, 2020

- **Model the lexicon**: A straightforward basis for segmentation is to build a lexicon. One of the lexicon-building algorithms, Byte pair encoding (BPE) (Sennrich et al., 2016), is popular for NLP preprocessing. It iteratively searches for the most common n-gram pairs and adds them into the n-gram lexicon. Some other models such as the Chunk-Based Learner (McCauley and Christiansen, 2019) and PARSER (Perruchet and Vinter, 1998) are also based on the local statistics of tokens (e.g., token frequency, mutual information, or transitional probability).
- **Model the grammar**: Some studies attempted to analyze the grammar patterns of sentences and then parse/segment the sentences based on these patterns. To find the optimal grammar, de Marcken (1996) used Minimum Description Length, and Johnson and Goldwater (2009) used the Hierarchical Dirichlet Process.
- **Model the sequences**: Recurrent neural networks and its variations are able to learn the sequential patterns in language and to perform text segmentation (Chung et al., 2017; Kawakami et al., 2019; Sun and Deng, 2018; Zhikov et al., 2013).

In general, lexicon models capture only the local statistics of the tokens so they tend to be short-sighted at the global level (e.g. long-distance dependencies). The other two types of models, in contrast, learn how the tokens co-occur globally. Yet, the ways grammar models and sequence models learn the global information makes them more complicated and computing-intensive than the lexicon models.

In this paper we propose a model that builds a lexicon, but does so by using both local and global information. Our model is not only a computational model but also a cognitive model: it is inspired by cognitive phenomena, and it needs only basic and light-weight computations which makes it cognitively more plausible than the grammar- and sequence-learning models mentioned above. We show that our model can effectively detect the cognitive units in language with an efficient procedure. We also show that our model can detect linguistically meaningful units. We further evaluate our model on traditional word segmentation tasks.

2 The Less-is-better Model

2.1 Cognitive principles

We want our system to mimic human cognitive processes of language unitization/segmentation by simulating not only the behavioral output, but also the cognitive mechanism. We designed such a computational model by emulating three cognitive phenomena: the principle of least effort, larger-first processing, and passive and active forgetting.

The principle of least effort: The essence of the model is a simple and natural cognitive principle: the principle of least effort (Zipf, 1949), which says human cognition and behavior are economic; they prefer to spend the least effort or resources to obtain the largest reward. Since a language sequence can be segmented into different sequences of language chunks, we assume the cognitive units are the language chunks in the sequence which follow the principle of least effort.

Larger-first processing: As we mentioned, any language chunk may be the cognitive unit, short or long. A broadly known finding is that global/larger processing has priority over local/smaller processing for visual scene recognition; an effect named "global precedence" (Navon, 1977). This follows from the principle of least effort: the larger the units we process, the fewer processing steps we need to take. For visual word processing, the word superiority effect (Reicher, 1969) shows the precedence of words over recognizing letters. Recent work (Snell and Grainger, 2017; Yang et al., 2020) extends global precedence to the level beyond words, and also shows that we do not process only the larger units: smaller units also have a chance to become the processing units when processing larger units does not aid comprehension. In other words, cognitive units may be of any size, but the larger have priority.

Passive and active forgetting: To mimic human cognition, the model should have a flexible memory to store and update information. Forgetting is critical to prevent the accumulation of an extremely large

number of memory engrams. It has been commonly held that forgetting is merely the passive decay of the memory engram over time, but recent studies put forward that forgetting can also be an active process (Davis and Zhong, 2017; Gravitz, 2019). Passive forgetting by decay can clean up the engrams that are no longer used in our brains. However, our brains may sometimes need to suppress counter-productive engrams immediately. Active forgetting may thus be called upon to eliminate the unwanted engram's memory traces, which enhances the memory management system (Davis and Zhong, 2017; Oehrn et al., 2018).

2.2 General idea

We assume the cognitive units are the chunks in the language sequence which follow the principle of least effort (Section 2.1). In other words, the less information we need to encode the language material, the better cognitive units we have. This less-is-better assumption grounds our model, so we named it Less-is-Better, or LiB for short.

The LiB model accepts any sequence S of atomic symbols s: $S = (s_1, s_2, \ldots)$, as the input. A collection of S forms a document D and all D together form the training corpus. S can be segmented into chunk tokens $(c_1, \ldots, c_N)$, where each chunk is a subsequence of S: $c = (s_i, \ldots, s_j)$ and N is the number of chunk tokens in S. The segmentation is based on a lexicon L (Fig. 1) where all chunk types are stored in order. The ordinal number of chunk type c in L is denoted $\Theta(c)$, and $|L|$ is the number of chunk types in L.

Let $I(c)$ be the amount of information (the number of encoding bits) required to identify each chunk type in L, that is, $I(c) = \log_2 |L|$, and $I(S)$ be the amount of information required for the input S, then: $I(S) = I(c)N$. Our model aims to minimize the expected encoding information to extract the cognitive units in any S, which means minimizing $E[I(S)]$, which is accomplished by simultaneously reducing $|L|$ (smaller $|L|$ means lower $I(c)$) and $E[N]$ (the expected number of chunk tokens in S). In practice our model:

1. Starts with an empty L;
2. Randomly selects a D from the corpus and analyzes the S in D;
3. Adds previously unseen symbols s as (atomic) chunk types to L;
4. Recursively combines adjacent chunk tokens into new chunk types, reducing $E[N]$ but increasing $|L|$;
5. Removes less useful types from L, reducing $|L|$;
6. Repeats steps 2 to 5 for a predetermined number of epochs.

The LiB model can segment any string S into a sequence of chunks $(c_1, ..., c_N)$ based on the lexicon L. The chunk types in L are ordered based on their importance inferred from the segmentation. The lexicon quality and the segmentation result mutually affect each other: LiB learns from its own segmentation results and updates L accordingly, then improves its next segmentation (Figure 1). The bootstrap procedure makes the model unsupervised.

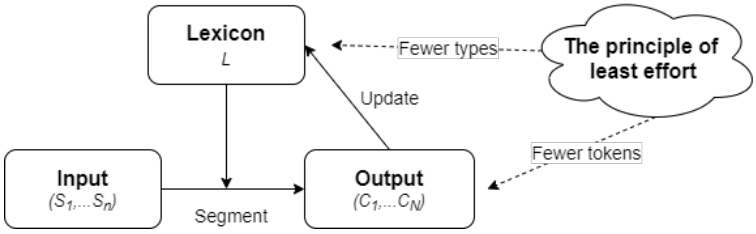

Figure 1: Information flow in the LiB model.

2.3 Implementation

2.3.1 Segmentation

Larger-first selection: An S can be segmented in different ways. For example, if both "going" and "goingto" are in L, and the given S is "goingtorain", then the first chunk token can be "going" or "goingto". The Larger-first principle (Section 2.1) dictates that LiB takes the largest substring of S that matches a chunk type in L (in the example case, it is "goingto"), i.e. greedy matching, and selects it as a chunk token (segment). If there is no chunk type in L that matches the current S, the first symbol s becomes the selected chunk token.

Chunk evaluation: In most cases, selecting larger chunk tokens will reduce the number of tokens N in S, but in some cases it will not. Let us continue the example we gave: If "goingtor", "a", "in", and "rain" are also in L, the largest chunk token becomes "goingtor", resulting in the segmentation "goingtor/a/in". If "goingto" had been selected, this would result in "goingto/rain". Hence, selecting the largest chunk type resulted in a larger N. The average chunk token sizes of the two segmentations are 5.5 and 3.6 letters, respectively.

In order to test whether the selected chunk type c reduces N, LiB compares the proposed segmentation to the segmentation that results if c is not in L, i.e., if the second largest chunk type in L is selected instead of c. In case L cannot provide a second largest chunk token, there is no evaluation and c is selected directly. Otherwise, c is evaluated as "Good" if it results in fewer chunk tokens or in the same number of tokens but with lower total ordinal numbers (i.e., chunks that are higher up in the lexicon):

$$\texttt{segment}(S, L) : S \rightarrow (c_1, c_2, \ldots, c_N)$$
$$\texttt{segment}(S, L - c) : S \rightarrow (c'_1, c'_2, \ldots, c'_{N'})$$
$$\texttt{evaluate}(c) = \begin{cases} \begin{cases} \text{Good} & \text{if } N < N' \\ \text{Bad} & \text{if } N > N' \end{cases} & \text{if } N \neq N' \\ \begin{cases} \text{Good} & \text{if } \sum_{i=1}^{N} \Theta(c_i) \leq \sum_{i=1}^{N'} \Theta(c'_i) \\ \text{Bad} & \text{if } \sum_{i=1}^{N} \Theta(c_i) > \sum_{i=1}^{N'} \Theta(c'_i) \end{cases} & \text{if } N = N' \end{cases}$$

If $\texttt{evaluate}(c)$ is Good, c is selected; otherwise, the second largest chunk token is selected.

2.3.2 Lexicon update

Memorizing: LiB learns new chunks from the segmentation results. There are two types of new chunks in the results: unknown symbols $s \notin L$ and concatenations of known chunks (c_i, c_{i+1}) (with $c_i \in L$ and $c_{i+1} \in L$) that occur consecutively in S. L starts empty, learns the symbol chunks, then the smallest chunks construct larger chunks and the larger chunks construct even larger chunks. Thus, L can contain chunks in different sizes.

The number of all (c_i, c_{i+1}) in the training corpus can be enormous, and most of them are infrequent chunks. In order to reduce the lexicon size $|L|$, LiB will memorize all s, but not all (c_i, c_{i+1}). To recognize the frequent chunks, a strategy is to count all chunks' occurrences and delete the infrequent ones (Perruchet and Vinter, 1998). However, this strategy asks for storing all chunks at the beginning, which is memory inefficient for both a brain and a computer. Thus, LiB adopts a sampling strategy: The model samples from all possible (c_i, c_{i+1}) tokens in the current S and memorizes only the tokens which were sampled at least twice. The probability of sampling a chunk pair is the sampling probability α. The sampling strategy is implicitly sensitive to the chunk token frequency in the text. It makes sure that even without explicit counting, higher-frequency chunks have a higher probability to be memorized. The at-least-twice strategy is not cognitively inspired but heuristic; it helps to prevent memorization of many arbitrary chunks.

Re-ranking and active forgetting: To avoid storing the frequencies of all possible chunk types, and to be more efficient, LiB bypasses explicit frequency counting of chunk types. Instead, LiB encodes the types' importance by their ordinals $\Theta(c)$ in L – the lower the more important. The importance reflects not only the frequency but also the principle of least effort (preference for fewer tokens and fewer types). In general, newly memorized chunk types are less frequent than known chunk types, so new chunk types are appended to the tail of L. The ordinals of known chunk types also need to be adjusted after new training text data comes in. The chunk evaluation we described in Section 2.3.1 is not only for segmentation, but also for importance re-ranking. The "good" chunk types, which result in fewer chunk tokens in S, will move closer to the lexicon head (i.e., lower ordinal); The "bad" chunk types, which result in more chunk tokens in S, will move closer to the lexicon tail, i.e., they get a higher ordinal number. The updated $\Theta(c)$ of a chunk type is relative to its previous ordinal $\Theta'(c)$ in L:

$$\Theta(c) = \begin{cases} \lfloor \Theta'(c)(1-\Delta) \rfloor & \text{if } c \text{ is good} \\ \lfloor \Theta'(c)(1+\Delta) \rfloor & \text{if } c \text{ is bad} \end{cases}$$

where $0 < \Delta < 1$ is the re-ranking rate. In case the updated $\Theta(c) > |L|$, c will be deleted from L.

Passive forgetting: Obviously, the re-ranking also influences other chunk types whose ordinals are between $\Theta(c)$ and $\Theta'(c)$. So even though the sampling strategy of the memorizer may add a few infrequent chunk types into L, the re-ranker will move them closer to the tail of L. Those chunk types, as well as the "bad" chunk types, are "junk chunks" which increase $I(c)$. The passive forgetter removes them from L to reduce $I(c)$.

The junk chunk types tend to be at the tail of L, but the tail may also store some non-junk types. A cognitive strategy to avoid deleting them is *waiting* for more evidence. So instead of deleting these types immediately, LiB uses a soft deleting strategy: after each training epoch, LiB will select the last $\omega|L|$ (at least one) chunk types in L and assign them a probation period τ. Here, ω is the forgetting ratio and τ is the remaining time until deletion; it is initialized at τ_0 and decreases by one after each training epoch (LiB analyzes one document D in each training epoch). Once the probation time is over, when $\tau = 0$, the chunk is forgotten (i.e., removed from L). If a chunk type was evaluated as "good" during its probation period, its probation is cancelled. The c that occur in fewer documents are more likely to be forgotten.

3 Model Training

We trained the LiB model on both English and Chinese materials (Table 1). The English material is **BR-phono**, which is a branch of the Brent corpus (Bernstein-Ratner, 1987), containing phonetic transcriptions of utterances directed at children. We used it for testing segmentation of spoken language. LiB accepts the document as an input batch in each training epoch but the utterances in the BR-phono corpus have no document boundaries. We randomly sampled 200 utterances (without replacement) from BR-phono to form one document and repeated this 400 times to create 400 documents for model training. The Chinese materials are taken from Chinese Treebank 8.0 (**CTB8**) (Xue et al., 2013), which is a hybrid-domain corpus (news reports, government documents, magazine articles, conversations, web discussions, and weblogs). As preprocessing, we replaced all the Roman letters and Arabic numbers with [*X*], and regarded all punctuation as sequence boundaries.

In order to examine the unsupervised performance of LiB, all spaces in the corpora were removed before training. We trained LiB on BR-phono and on CTB8 separately. The parameter settings are shown in Appendix A. The example segmentations with increasing number of training epochs are shown in Appendix B. The related code and preprocessed corpora are available online[1].

[1] `https://github.com/ray306/LiB`

Corpus	Documents	Sentences	Word tokens	Word types
BR-phono	400	9,790	33,399	1,321
CTB8	3,007	236,132	1,376,142	65,410
MSR	/	18,236	89,917	11,728
PKU	/	15,492	88,327	12,422

Table 1: The training and test corpus statistics after preprocessing. MSR and PKU are the (Chinese) test corpora which are mentioned in Section 4.5. Word units are presegmented in the CTB8, MSR, and PKU corpora.

4 Model Evaluation

4.1 Subchunks

After training, we evaluated the chunk units in the training corpora from two information-theoretical views that bear a relation to cognitive processing: description length and language model surprisal. We also examined the performance of LiB on word segmentation tasks. However, since LiB can learn new chunks from the concatenation of known chunks, the learned chunks are not only words, but also possible multi-word expressions. For the word segmentation task, we want to know the words in those multi-word expressions, so we had LiB find the subchunks $c^\flat$, which are the chunks inside the original chunks (e.g., "you" and "are" inside "youare"), and regarded the subchunks as the words. LiB defines the subchunk by searching all the potential chunk sequences in the original chunk (c_{raw}) and selecting the sequence with lowest sum of ordinals unless c_{raw} has the lowest sum:

$$(c_1^\flat, \ldots, c_n^\flat) = \underset{(c_1,\ldots,c_n)}{\arg\min} \left(\sum_i \Theta(c_i) \right), \text{where } (c_1, \ldots, c_n) = c_{raw}$$

$$\text{Subchunk(s) of } c_{raw} = \begin{cases} (c_1^\flat, \ldots, c_n^\flat) & \text{if } \max_i(\Theta(c_i^\flat)) < \Theta(c_{raw}) \\ c_{raw} & \text{otherwise} \end{cases}$$

4.2 Qualitative evaluation

Since the LiB lexicon is ordered, we may examine the head of the trained lexicons (Table 9), which are the highest-ranked chunk units. They show that LiB appears to learn common words and collocations. Among the learned units we observe some collocations (e.g., "that'sa") which are not linguistic phrases. The lexicon of LiB trained on CTB8 shows that the high-ranked Chinese chunk units are usually bigrams (Appendix C). The middle and the tail of the trained lexicons are also shown in Appendix C. We present examples of chunk and subchunk segmentation results in Table 3. The results show the chunk units include common collocations, while the subchunk units are very close to the linguistic words.

4.3 Description length evaluation

LiB provides two types of new units to segment language: **LiB chunks** are the raw segmentation result of LiB, and **LiB subchunks** are the subchunks inside LiB chunks. In order to examine the encoding efficiency of LiB chunks and LiB subchunks, we compared the description lengths (DL) on different segmentations. The DL is the number of bits required to represent the corpus; it sums the number of bits required to encode the lexicon and the number of bits required to encode the corpus when segmented by the lexicon (Zhikov et al., 2013):

$$DL(total) = DL(lexicon) + DL(corpus) = -\sum_{i=1}^{\#s} Freq(s_i) \log_2 P(s_i) - \sum_{j=1}^{\#u} Freq(u_j) \log_2 P(u_j)$$

Corpus	Top 50 entries (translated) in Lexicon
BRphono	the, yeah, you, what, wanna, can you, two, and, that's, okay, four, now, it, they're, he's, in, look, with, you want, who, he, that, all, your, here, i think, put, that's a, what's, you can, his, my, see, you wanna, no, is that, high, whose, this, good, there's, very, see the, its a, is it, alright, this is, are you, ing, have
CTB8	haven't, China, we, economics, already, kid, but, education, can, now, government, country, a, these, self, can't, if, journalist, today, they, although, require, tech, process, this, Xinhua News Agency, wish, issue, is, mainland, because, some, and, all are, so, now, may, Taiwan, should, political, development, also is, also is, society, such, via, continue, isn't, Shanghai, 's

Table 2: Transliterations/translations into English of the top 50 entries in the lexicons. The original results of BRphono are in phonemic characters, and the original results of CTB8 are the Chinese characters. For completeness, in Appendix C we repeat these results with the original results included.

Corpus	Level	Segmentation
BRphono	Input	allrightwhydon'tweputhimawaynow
	Chunks	allright·whydon't·we·puthimaway·now
	Subchunks	all·right·why·don't·we·put·him·away·now
	Words	all·right·why·don't·we·put·him·away·now
CTB8	Input	这个出口信贷项目委托中国银行为代理银行
	Chunks	这个·出口信贷·项目·委托·中国银行·为·代理·银行
	Subchunks	这个·出口·信贷·项目·委托·中国·银行·为·代理·银行
	Words	这·个·出口·信贷·项目·委托·中国·银行·为·代理·银行

Table 3: Example segmentations of strings in the two corpora. BRphono's results are transcribed into English words for ease of presentation.

Here, $\#s$ denotes the number of unique symbols s in L (either as a single-symbol chunk or as part of a larger chunk); *Freq*(s_i) and $P(s_i)$ are the occurrence count and ratio of s_i in L; $\#u$ denotes the number of unique units u in the corpus; *Freq*(u_j) and $P(u_j)$ are the occurrence count and ratio of u_j in the corpus.

As benchmarks, we used **Symbol** (the indivisible units; in our two corpora, phonemes and characters respectively), **Word** (the words presegmented in the corpora), and **BPE subword** (the Byte Pair generated by SentencePiece (Kudo and Richardson, 2018) with default parameters setting). The DL result (Table 4) shows that LiB chunks result in shortest DL; they minimze the information; they are the most concise encodings.

4.4 Language model evaluation

Besides the DL, which compares the information efficiencies of different lexicons, we are also interested in whether the LiB lexicon can reflect the mental lexicon. We lack a ground truth of what is in the putative mental lexicon. However, we can regard natural language material as a large-scale result of human language use and language behavior. Trained on a very large corpus, a recent study by Brown et al. (2020) shows that Language Models (LMs) can closely predict human performance on various language tasks. LMs capture the probabilistic constraints in natural language and perform the tasks by making predictions, which is a fundamental cognitive function (Bar, 2007). So, by measuring the prediction surprisal in the corpus segmented by different lexicons, we can evaluate different lexicons from a cognitive view, and we presume that the lexicon that gets the best LM performance is a better approximation of the mental lexicon.

Many studies have shown that word surprisal is positively correlated with human word-reading time (Monsalve et al., 2012; Smith and Levy, 2013) and size of the N400 component in EEG (Frank et al., 2015). From the cognitive principle of least effort, it follows that readers try to minimize reading time.

		Segmentation				
Corpus	**Evaluation metric**	**Symbol**	**BPE subword**	**Word**	**LiB subchunk**	**LiB chunk**
BRphono	Average length	1	2.8	2.9	2.9	3.6
	Lexicon size	50	5,574	1,321	1,119	1,869
	DL(lexicon)	**<1**	173	28	24	47
	DL(corpus)	490	278	262	258	**233**
	DL(total)	490	451	289	282	**281**
CTB8	Average length	1	1.4	1.7	1.7	1.9
	Lexicon size	4,697	7,980	65,410	24,763	39,320
	DL(lexicon)	**57**	133	1,767	621	1,153
	DL(corpus)	21,864	18,229	15,669	16,188	**15,602**
	DL(total)	21,921	18,362	17,436	16,809	**16,755**

Table 4: Average token lengths, lexicon sizes, and the DL results of different types of segmentation on the two corpora. The unit of Average Length is phoneme (BRphono) or Chinese character (CTB8). The unit of DL is kilobit.

		Segmentation				
Corpus	**Model**	**Symbol**	**BPE subword**	**Word**	**LiB subchunk**	**LiB chunk**
BRphono	2-gram	1.539	0.695	0.677	0.649	**0.548**
	3-gram	0.950	0.390	0.405	0.378	**0.335**
CTB8	2-gram	2.466	1.932	1.617	1.668	**1.452**
	3-gram	1.404	0.827	0.806	0.748	**0.626**

Table 5: Bits-per-character scores on different segmentations.

Hence, it follows that readers would try to find lexical units such that total surprisal is also minimized.

Surprisal, defined as $-\log_2(P(w|\text{context}))$, is not comparable between models with different segmentations. Instead we use bits per character (BPC) (Graves, 2013), which is average surprisal$/|c|$, where $|c|$ is the average chunk length over the whole test set. We tested the segmentations[2] on both bigram and trigram language models and the results show that the corpora represented by LiB chunks achieve the lowest surprisal (Table 5).

4.5 Word segmentation evaluation

As we already illustrated in Table 3, subchunk units tend to be close to linguistic words. We thus tested LiB subchunks as a resource for word segmentation. To evaluate LiB on English word segmentation, we compared LiB with Adaptor Grammar (AG) (Johnson and Goldwater, 2009), which achieves state-of-the-art performance on the segmentation task of BR-phono. AG requires grammar construction rules that encode prior linguistic knowledge. These rules presuppose knowledge about unigrams only, or unigrams+collocations, or unigrams+collocations+syllables. This yields three versions of AG. Table 6a shows that AG(syllable), whose rules carry extra linguistic knowledge (Johnson and Goldwater, 2009), achieves the highest score. The score of LiB is higher than AG(unigram) and slightly lower than AG(collocations), the two versions of AG comparable to our approach. AG(syllable) presumes knowledge that our model does not have (and that could possibly benefit LiB).

In the Chinese segmentation task. we compared LiB with three popular word segmentation toolboxes: Jieba[3], THULAC (Sun et al., 2016), and pkuseg (Luo et al., 2019). These toolboxes are supervised, learning the ground truth (word boundaries) during training. For comparison, we also modified a su-

[2]The code of the BPC calculations was modified from a Github project: `https://github.com/joshualoehr/ngram-language-model`. We kept all tokens during training.

[3]`https://github.com/fxsjy/jieba`

pervised LiB (LiB(sup)) for the word segmentation task. LiB(sup) skips the training phase. Instead, it counts all the ground-truth words in the training set and adds them as the chunk types to L. The higher the frequency of a type in the training set, the smaller its ordinal in L. We trained and tested the models on CTB8. To test the generalization performance of the models in the word segmentation task, we also test the training result on two additional corpora: MSR and PKU (Table 1) provided by the Second International Chinese Word Segmentation Bakeoff (Emerson, 2005). The segmentation rules are slightly different among MSR, PKU, and CTB8. MSR and PKU are news domain, which is different from CTB8. MSR and PKU were preprocessed in the same way as CTB8.

Table 6b shows that the scores of the unsupervised original version of LiB are lower than the supervised models[4], but the scores of the supervised version of LiB are close to the supervised models and are even higher on MSR. Due to the low out-of-vocabulary (OOV) rate of MSR (Emerson, 2005), the good performance on MSR shows that the lexicon is important for LiB. The only difference between the two versions of LiB is in their lexicons: the original LiB learned the lexicon from zero and the supervised LiB directly uses the ground-truth words in its lexicon. It shows that the segmentation module in LiB is appropriate for the word segmentation task.

[a]

Model	Scores
AG (unigram)	56
AG (collocations)	76
AG (syllable)	**87**
LiB subchunk	71

[b]

	Test set scores		
Model	**CTB8**	**MSR**	**PKU**
Jieba	87.1	82.8	87.1
THULAC	94.6	83.5	89.1
pkuseg	**95.7**	83.7	**89.7**
LiB subchunk	76.1	78.7	78.9
LiB(sup) chunk	94.7	**84.5**	88.3

Table 6: Token F1 scores (%) of segmentations. [a] the scores on BR-phono by three versions of Adaptor Grammar (AG) and LiB subchunks. [b] the scores of Jieba, THULAC, PKUSEG, LiB subchunks, and LiB(sup) chunks. LiB(sup) represents the supervised adaptation of LiB.

5 Conclusions and Future Work

This paper presented an unsupervised model, LiB, to simulate the human cognitive process of language unitization/segmentation. Following the principles of least effort, larger-first processing, and passive and active forgetting, LiB incrementally builds a lexicon which can minimize the number of unit tokens (alleviating the effort of analysis) and unit types (alleviating the effort of storage) at the same time on any given corpus. Moreover, it is able to segment the corpus, or any other text in the same language, based on the induced lexicon. The computations in LiB are light-weight, which makes it very efficient. The LiB-generated lexicon shows optimal performances among different types of lexicons (e.g., ground-truth words) both in terms of description length and in terms of statistical language model surprisal, both of which are associated with cognitive processing. The workflow design and the computation requirement make LiB cognitively plausible, and the results suggest that the LiB lexicon may be a useful proxy of the mental lexicon.

Future work will be to allow skip-gram units in the lexicon. Skip-grams may help to capture longer-distance dependencies, and further lessen the cognitive effort by reducing the number of unit types/tokens. Furthermore, as the word segmentation results of the current LiB are not ideal, we hypothesize that skip-gram units may also benefit the detection of infrequent named entities (e.g., the skip-gram "Mr._said" helps to detect "Mortimer" in "Mr.Mortimersaid") and thus improve the word segmentation performance. Other future work includes a LiB variant that accepts speech input and a semi-supervised LiB variant that uses semantic knowledge (e.g., word embeddings) to enhance the language unitization.

[4]The scores of Jieba, THULAC, and pkuseg are provided by `https://github.com/lancopku/pkuseg-python`

References

Inbal Arnon and Neal Snider. 2010. More than words: Frequency effects for multi-word phrases. *J. Mem. Lang.*, 62(1):67–82.

Colin Bannard and Danielle Matthews. 2008. Stored word sequences in language learning: the effect of familiarity on children's repetition of four-word combinations. *Psychol. Sci.*, 19(3):241–248, March.

Moshe Bar. 2007. The proactive brain: using analogies and associations to generate predictions. *Trends Cogn. Sci.*, 11(7):280–289.

Nan Bernstein-Ratner. 1987. The phonology of parent-child speech. *Children's language*, 6(3).

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are Few-Shot learners. *arXiv preprint arXiv:2005.14165*, May.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April. OpenReview.net.

Ronald L Davis and Yi Zhong. 2017. The biology of forgetting — a perspective. *Neuron*, 95(3):490–503, August.

Carl de Marcken. 1996. *Unsupervised language acquisition*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language processing*.

Robert Fiorentino, Yuka Naito-Billen, Jamie Bost, and Ella Fund-Reznicek. 2014. Electrophysiological evidence for the morpheme-based combinatoric processing of English compounds. *Cogn. Neuropsychol.*, 31(1-2):123–146.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain Lang.*, 140:1–11, January.

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

Lauren Gravitz. 2019. The importance of forgetting. *Nature*, 571:S12–S14.

Ray Jackendoff. 2002. What's in the lexicon? In Sieb Nooteboom, Fred Weerman, and Frank Wijnen, editors, *Storage and Computation in the Language Faculty*, pages 23–58. Springer Netherlands, Dordrecht.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2019. Learning to discover, ground and use words with segmental neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6441, Stroudsburg, PA, USA. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. PKUSEG: A toolkit for multi-domain Chinese word segmentation. *arXiv preprint arXiv:1906.11455*, June.

Lucy J MacGregor and Yury Shtyrov. 2013. Multiple routes for compound word processing in the brain: Evidence from EEG. *Brain Lang.*, 126(2):217–229.

Stewart M McCauley and Morten H Christiansen. 2019. Language learning as language use: A cross-linguistic model of child language development. *Psychol. Rev.*, 126(1):1–51, January.

Irene Fernandez Monsalve, Stefan L Frank, and Gabriella Vigliocco. 2012. Lexical surprisal as a general predictor of reading time. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 398–408.

David Navon. 1977. Forest before trees: The precedence of global features in visual perception. *Cogn. Psychol.*, 9(3):353–383.

Carina R Oehrn, Juergen Fell, Conrad Baumann, Timm Rosburg, Eva Ludowig, Henrik Kessler, Simon Hanslmayr, and Nikolai Axmacher. 2018. Direct electrophysiological evidence for prefrontal control of hippocampal processing during voluntary forgetting. *Curr. Biol.*, 28(18):3016–3022.e4, September.

Pierre Perruchet and Annie Vinter. 1998. PARSER: A model for word segmentation. *J. Mem. Lang.*, 39(2):246–263, August.

Gerald M. Reicher. 1969. Perceptual recognition as a function of meaningfulness of stimulus material. *J. Exp. Psychol.*, 81(2):275–280, August.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319, September.

Joshua Snell and Jonathan Grainger. 2017. The sentence superiority effect revisited. *Cognition*, 168:217–221, November.

Zhiqing Sun and Zhi-Hong Deng. 2018. Unsupervised neural word segmentation for Chinese via segmental language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4915–4920, Brussels, Belgium. Association for Computational Linguistics.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. Thulac: An efficient lexical analyzer for Chinese.

Marcus Taft. 2013. *Reading and the mental lexicon*. Psychology Press.

Nianwen Xue, Xiuhong Zhang, Zixin Jiang, Martha Palmer, Fei Xia, Fu-Dong Chiou, and Meiyu Chang. 2013. Chinese treebank 8.0 LDC2013T21. *Linguistic Data Consortium, Philadelphia*.

Jinbiao Yang, Qing Cai, and Xing Tian. 2020. How do we segment text? two-stage chunking operation in reading. *eNeuro*, May.

Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. 2013. An efficient algorithm for unsupervised word segmentation with branching entropy and MDL. *Information and Media Technologies*, 8(2):514–527.

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort*, volume 573. Addison-Wesley Press, Oxford, England.

A Training parameter settings

Since BR-phono is a child-directed speech corpus, its chunk types are usually very common, and so they often have much higher document ratios than CTB8 chunks. We use a lower τ_0, which is related to document ratio, to balance the corpus difference. The number of training epochs for CTB8, which is large-scale, was set to a higher number than for BR-phono. The epochs numbers are well beyond the convergence points. α and Δ mainly affect the training speed, while ω and τ_0 mainly affect $|L|$. The current parameter settings may not be optimal for end tasks such as word segmentation; in preliminary experiments we optimized for speed[5].

Corpus	α	Δ	ω	τ_0	**epochs**
BR-phono	0.25	0.2	0.0001	10	5,000
CTB8				500	50,000

Table 7: The parameter settings in the training on two corpora. α is the sampling probability, Δ the re-ranking rate, ω the forgetting ratio, τ_0 the probation period.

B Segmentations with increasing number of training epochs

The progression in chunking over training epochs before convergence (Table 8) shows LiB can learn some word chunks even in the very early epochs. Also, Table 8 illustrates that convergence is reached well before the preset number of runs.

Corpus	**Epoch**	**Segmentation**
BRphono	0	Olr9tW9dontwipUthIm6wenQ
	1	O·l·r·9·t·W·9·don·t·w·i·pUt·h·I·m·6·w·e·nQ
	2	Ol·r·9t·W·9·dont·wi·pUt·h·I·m·6·we·nQ
	10	Olr9t·W9·dont·wi·pUt·hIm·6we·nQ
	100	Olr9t·W·9dont·wi·pUthIm6we·nQ
	1,000	Olr9t·W·9dont·wi·pUthIm6we·nQ
CTB8	0	这个出口信贷项目委托中国银行为代理银行
	1	这·个·出·口·信·贷·项·目·委·托·中·国·银·行·为·代·理·银·行
	2	这·个·出·口·信·贷·项·目·委·托·中国·银·行·为·代·理·银·行
	10	这·个·出口·信·贷·项·目·委·托·中国·银·行·为·代·理·银·行
	100	这个·出口·信·贷·项目·委·托·中国·银·行·为·代·理·银·行
	1,000	这个·出口·信贷·项目·委·托·中国·银行·为·代·理·银行
	10,000	这个·出口信贷·项目·委托·中国银行·为·代理·银行

Table 8: Example segmentations of strings in the two corpora with increasing number of training epochs. See Table 3 for the correct word-level segementation.

[5]The training of BR-phone costs 57 s and the training of CTB8 costs 31 min 55 s. The code is written in pure Python 3.7 and ran on a single core of Intel Core i5-7300HQ.

C Top, middle and tail entries in lexicon

Corpus	Entries in Lexicon
BRphono (Top 50)	D6 **the**, y& **yeah**, yu **you**, WAt **what**, wan6 **wanna**, k&nyu **can you**, tu **two**, &nd **and**, D&ts **that's**, oke **okay**, f% **four**, nQ **now**, It **it**, D* **they're**, hiz **he's**, In **in**, lUk **look**, wIT **with**, yuwant **you want**, hu **who**, hi **he**, D&t **that**, Ol **all**, y) **your**, h(**here**, 9TINk **i think**, pUt **put**, D&ts6 **that's a**, WAts **what's**, yuk&n **you can**, hIz **his**, m9 **my**, si **see**, yuwan6 **you wanna**, no **no**, IzD&t **is that**, h9 **high**, huz **whose**, DIs **this**, gUd **good**, D*z **there's**, v*i **very**, siD6 **see the**, Its6 **its a**, IzIt **is it**, Olr9t **alright**, DIsIz **this is**, #yu **are you**, IN **ing**, h&v **have**
BRphono (Middle 20)	siD&t **see that**, nik, lEtmiQt **let me out**, DIsgoz **this goes**, d&diznat **daddy's not**, 9ms%i **i'm sorry**, kIN, lUksl9k6n9s, wITDiz **with these**, hizwe **he's way**, lON **long**, h&p **happen**, lEtssiIf **let's see if**, lEtspUthIm6we **let's put him away**, diIzf%, pR, brEkf6st **breakfast**, h9c* **high chair**, lUk&tD6bUk **look at the book**, W*zD6kIti
BRphono (Tail 20)	Nkyu, T, uyuwant, * **air**, 3, (**ear**, Z, c,), M, InhIzhQs, 6mily6 **amelia**, dOghQs **doghouse**, wITt7z **with toys**, &ndsAmt9mzwi, holdh&ndz **hold hands**, tIkLmi **tickle me**, h9ke **high kay**, tekItQt, k&nyubrAShIzh*
CTB8 (Top 50)	没有 **haven't**, 中国 **China**, 我们 **we**, 经济 **economics**, 已经 **already**, 孩子 **kid**, 但是 **but**, 教育 **education**, 可以 **can**, 目前 **now**, 政府 **government**, 国家 **country**, 一个 **a**, 这些 **these**, 自己 **self**, 不能 **can't**, 如果 **if**, 记者 **journalist**, 今天 **today**, 他们 **they**, 虽然 **although**, 要求 **require**, 技术 **tech**, 进行 **process**, 这个 **this**, 新华社 **Xinhua News Agency**, 希望 **wish**, 问题 **issue**, 就是 **is**, 大陆 **mainland**, 因为 **because**, 一些 **some**, 以及 **and**, 都是 **all are**, 因此 **so**, 现在 **now**, 可能 **may**, 台湾 **Taiwan**, 应该 **should**, 政治 **political**, 发展 **development**, 也是 **also is**, 还是 **also is**, 社会 **society**, 这样 **such**, 通过 **via**, 继续 **continue**, 不是 **isn't**, 上海 **Shanghai**, 的 **'s**
CTB8 (Middle 20)	肝脏 **liver**, 军事政变推翻 **military coup overthrows**, 在其他地方 **in other places**, 在野势力 **opposition force**, 而且这个 **and this**, 泄的, 帮他 **help him**, 宝应县 **Baoying County**, 政治新闻 **political news**, 经济越 **economic more**, 塔肯, 迅速地 **rapidly**, 铅笔 **pencil**, 集体经济 **collective economy**, 起源 **origin**, 邓相扬协助 **Tang Xiangyang assisted**, 建制 **establishment**, 写完 **after writing**, 说的那样 **as said**, 后顾 **look back**
CTB8 (Tail 20)	存在主权 **there is sovereignty**, 确权 **confirm rights**, 草案还 **the draft also**, 桌会议, 第一首相 **the first prime minister**, 迪奥 **dior**, 长大了 **grown up**, 爱他 **love him**, 说他 **say him**, 子虚乌, 有没有参与 **did you participate**, 严谨的 **rigorous**, 仍然是 **is still**, 站上车, 运输署 **Transport Department**, 杀机 **murderous**, 决 **decided**, 建成通车 **completed and opened to traffic**, 主要嫌疑人赖昌星 **the main suspect Lai Changxing**, 已经向加拿大 **has to Canada**

Table 9: The top 50 entries, the middle 20 entries and the tail 20 entries in the lexicons. The original results of BRphono are in phonemic characters; we transcribed the entries containing complete words into English words (in bold font) for ease of presentation. The original results of CTB8 are the Chinese characters; we added the English translations (in bold font) with the entries containing complete words.