Human Cognition Based Word Segmentation Models

Shinjini Ghosh
Department of Electrical Engineering and Computer Science, MIT
shinghos@mit.edu

Raul Alcantara
Department of Electrical Engineering and Computer Science, MIT
ralcanta@mit.edu

Abstract

Segmentation of words from free speech or unsegmented text is an almost universally prevalent human skill. In this article, we build, implement and test three computational models of word segmentation based on human cognition—a probabilistic context-free grammar model, a probabilistic n-gram model with dynamic programming, and a statistical Viterbi algorithm based approach. We also investigate how they perform in comparison with human cognition experiments in similar conditions.

1 Introduction

Word segmentation is the process of determining the word boundaries in free-flowing speech or non-segmented text. Language learners of all ages are able to naturally demarcate word boundaries from continuous speech, even without appreciable pauses or other linguistic cues (as mentioned in Saffran et al. [7]). So how is that human cognition allows for word segmentation within such poverty of stimulus? And is there a way we can capture the same computationally for our use in language modeling and beyond?

There have been several attempts and ongoing work for finding different algorithms that will efficiently and accurately segment any given sentence, text, or speech, either in English or in another language. One of the pioneers in this was Saffran et al., who analyzed how children performed this task from a very young age. After her study, different approaches were taken, both in the classical statistics (Brent [1], Venkataraman [9]) and Bayesian realms (Goldwater et al. [2]). In this paper, we try to develop on and implement some of these methods, and analyze their performance on the segmentation task, given different unsegmented corpora, especially in relation with human judgements.

Final Project for Computational Cognitive Science, Fall 2020

2 Motivation

We believe that current state-of-the-art language models, which fail on natural language understanding and inference tasks, could benefit with human-inspired augmentations and that an improved word segmentation algorithm would further the current NLP frontiers in the capabilities of neural and non-neural language models, especially because most models currently in place crucially reply on segmenting words correctly. We wish to distil the knowledge gained from our understanding of human cognition into computational models and human-like intelligent systems.

3 Saffran Revisited, Computationally

Saffran et al. [7]'s groundbreaking paper delves into statistical learning by 8-month-old infants, and aims to probe one of the very basic human cognitive tasks, a fundamental task accomplished by almost every child in the world—segmentation of words from fluent speech. The authors state that 'successful' word segmentation by the infants, based on only 2 minutes of speech exposure, suggests that they have access to a powerful mechanism for computing the statistical properties of language input. This is a very important observation in building computational models of cognition regarding word segmentation, especially when coupled with the fact that there exists complex and widely varying acoustic structure of speech in different languages and hence, there is no invariant acoustic cue to word boundaries present in all languages.

In the class goal of 'reverse-engineering the human mind', Saffran et al.'s observations are crucial as we set out to use knowledge of how human intelligence works in order to build more human-like intelligence systems [Class Slides, Lecture 1]. As outlined in Goodman et al. [3], Piantadosi et al. [6], Piantadosi et al. [5] and multiple other papers, the probabilistic language of thought hypothesis believes that concepts have a language-like compositionality and encode probabilistic knowledge, thereupon relying on Bayesian inference for production. We also look at how Goldwater et al. [2] approach the word segmentation problem probabilistically, relying on word context. Extending from the word learning concepts of Lectures 1 and 5, and the categorization concepts of Lectures 22 and 23, we try to revisit Saffran et al.'s experiment, this time computationally.

3.1 Modeling with Probabilistic Context Free Grammars (PCFGs)

We use a Probabilistic Context Free Grammar to computationally explore Saffran et al.'s experiment, and see how our probabilistic model performs in comparison with human infants. In class, we saw how PCFGs perform Bayesian inference for the sentence 'I shot an elephant in my pajamas', and what the various valid parses for this sentence are. In Johnson et al. [4], Bayesian inference for PCFGs via MCMC is used for morphology. We adapt a similar ideology to word segmentation as follows, where a PCFG captures word segmentation as inference.

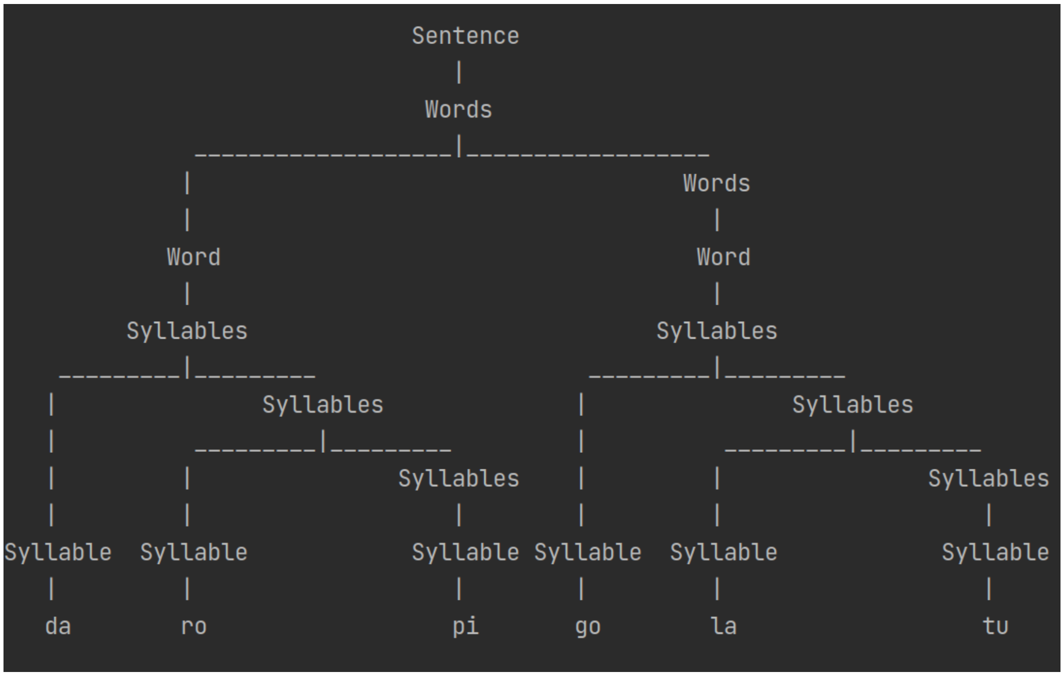
3.2 PCFG Setup

We modify the usual PCFG setup as follows. Instead of having an input sentence, we have an input speech stream, segmented into syllables. We assume that the smallest part of speech that infants can discern without external knowledge is syllables (Saffran et al. also take tri-syllabic words and look at probability transitions between word boundaries in infants), and our concern is how they break this syllable stream into words. We then segment the speech stream aka ‘Sentence’ into words, which further break into more words or a single word, which break into syllable(s).

Our sample PCFG thus looks as follows.

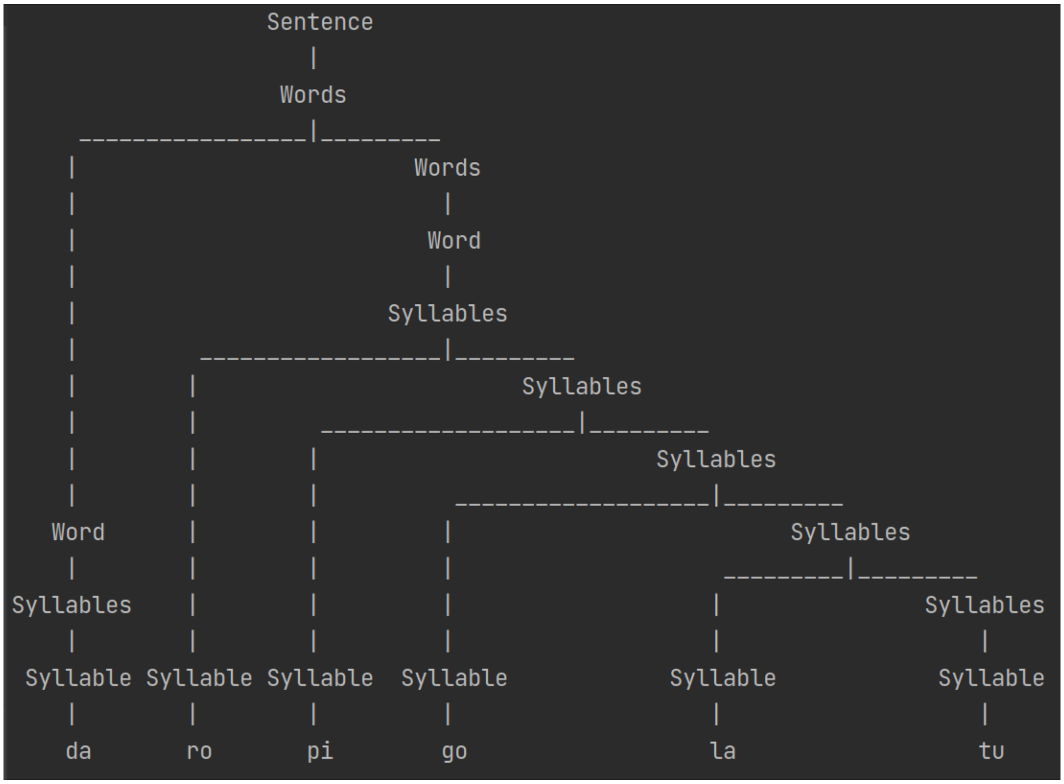
```
    """
    Sentence -> Words [1.0]
    Words -> Word Words [0.8] | Word [0.2]
    Word -> Syllables [1.0]
    Syllables -> Syllable Syllables [0.8] | Syllable [0.2]
    Syllable -> 'tu' [0.083]
    Syllable -> 'pi' [0.168]
    Syllable -> 'ro' [0.083]
    Syllable -> 'go' [0.083]
    Syllable -> 'la' [0.168]
    Syllable -> 'bu' [0.083]
    Syllable -> 'da' [0.083]
    Syllable -> 'ko' [0.083]
    Syllable -> 'ti' [0.083]
    Syllable -> 'du' [0.083]
    """
```

Just like Saffran et al., we generate speech stream by randomly concatenating words from the input vocabulary (of 2 minutes = 180 words). The syllable probabilities are then inferred from the speech stream, and the word/syllable break probabilities are a parameter that we tweak and see the results with. We then investigate the various word parses (and corresponding) that these PCFGs give us, as well as the probabilities of those parses. A sample parse tree with high probability is shown in Fig 1—it shows us how given a stream of syllables, our PCFG breaks down the input ‘da ro pi go la tu’ into two words ‘daropi’ and ‘golatu’. This is one of the “hard" input examples for the vocabulary consisting of the words “pigola", “golatu", and “daropi", because the ‘part-word’ “pigola" spanned the boundary between ‘daropi#golatu’. Below that, we have another parse tree in Fig 2—one with a low probability assigned by the parser, and clearly not adequate. We hypothesize that humans have access to such computing mechanism, and select a high probability parse tree to use in their daily lives.



```
                              Sentence
                                 |
                               Words
              ___________________|__________________
             |                                    Words
             |                                      |
            Word                                   Word
             |                                      |
         Syllables                              Syllables
    _________|_________                    _________|_________
   |               Syllables              |               Syllables
   |          _________|_________         |          _________|_________
   |         |               Syllables    |         |               Syllables
   |         |                   |        |         |                   |
Syllable  Syllable            Syllable Syllable  Syllable            Syllable
   |         |                   |        |         |                   |
   da        ro                  pi       go        la                  tu
```

Figure 1: Sample Parse Tree with High Probability

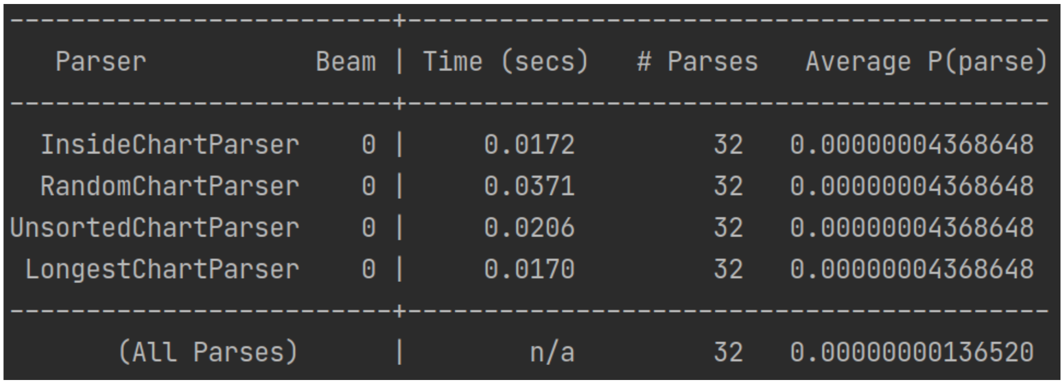


```
                   Sentence
                      |
                    Words
     _________________|_________
    |                         Words
    |                           |
    |                          Word
    |                           |
    |                       Syllables
    |         __________________|_________
    |        |                        Syllables
    |        |         ___________________|_________
    |        |        |                         Syllables
    |        |        |          ___________________|_________
   Word      |        |         |                         Syllables
    |        |        |         |                    _________|_________
Syllables    |        |         |                   |              Syllables
    |        |        |         |                   |                  |
 Syllable Syllable Syllable  Syllable            Syllable           Syllable
    |        |        |         |                   |                  |
    da       ro       pi        go                  la                 tu
```

Figure 2: Sample Parse Tree with Low Probability

3.3 Parser Ranking

We also try four different parsers and do a comparative analysis of the time taken by them to compute all parses of the input ‘da ro pi go la tu’, in an attempt to compare with human reaction times. We receive the following values, as shown in Fig 3. Saffran et al. gave the infants a much higher threshold of 2 seconds to judge word familiarity.



```
-------------------------+----------------------------------------
   Parser           Beam | Time (secs)   # Parses   Average P(parse)
-------------------------+----------------------------------------
  InsideChartParser    0 |      0.0172         32   0.00000004368648
  RandomChartParser    0 |      0.0371         32   0.00000004368648
UnsortedChartParser    0 |      0.0206         32   0.00000004368648
 LongestChartParser    0 |      0.0170         32   0.00000004368648
-------------------------+----------------------------------------
       (All Parses)      |         n/a         32   0.00000000136520
```

Figure 3: Parser Time Benchmarking

3.4 Results

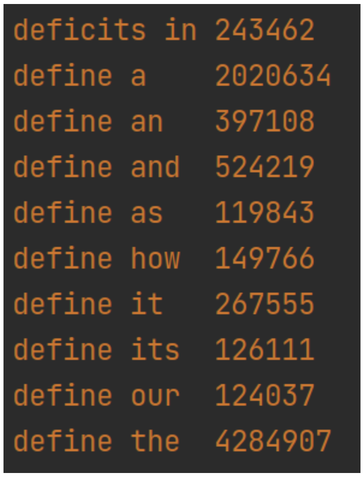
We see that just as Saffran et al. predicted for human infant judgements, for a majority of our randomized experiments, the PCFGs also pick up on a higher transitional probability between two sounds in a word, as compared to two sounds across word boundaries. We thus believe that a probabilistic model can be built to accurately pick up on cues similar to human cognition, while recognising the fact that this method works well only on smaller lexicons and as the number of words increase, the number of rules in a PCFG also increase drastically.

4 Dynamic Programming with Probabilistic n-gram Modeling

In addition to the PCFG modeling, we then go on to other probabilistic modeling techniques for word segmentation. We use a model implemented along the lines of Peter Norvig’s in the book ‘Beautiful Data’ by Segaran and Hammerbacher [8].

4.1 Dataset

We use Norvig’s pre-processed version of the Google Trillion Word Dataset distributed through the Linguistics Data Consortium. This dataset is trimmed of n-grams occurring lower than 40 times, unkified, and sentence demarcations are added. It readily gives us unigram and bigram probabilities, from which we can compute the conditional probabilities as well. A snapshot of the bigram counts data used is in Fig 4.



```
deficits in 243462
define a    2020634
define an   397108
define and  524219
define as   119843
define how  149766
define it   267555
define its  126111
define our  124037
define the  4284907
```

Figure 4: Bigram Counts Dataset Snapshot

4.2 Modeling and Implementation

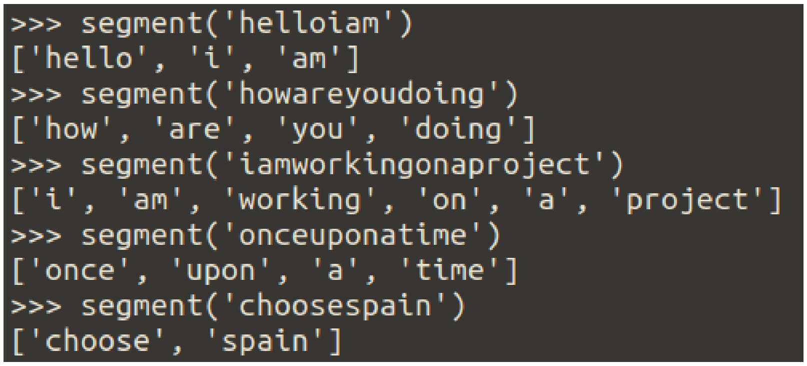
We use two probabilistic models—one based on unigrams and the other on bigrams. We recursively split a stream of text, computing the Naive Bayes probability of the sequence of words thus formed, and use dynamic programming to memoize our computation, preventing us from running into exponential times. The Bayes probability is computed using a probability distribution estimated from the counts in the pre-processed data files, and Laplace additive smoothing is used to estimate the probability of unknown words. We also use surprisal values for the bigram model. Finally, the segmentation with the highest probability, or the lowest surprisal, is chosen as our output segmentation.

4.3 Testing and Results

We create a unit test file, with segmentations of text stream, both straightforward and ambiguous, e.g., ‘choosespain’ can be segmented both as ‘choose spain’ or ‘chooses pain’. If we believe that humans use statistical inference, then we can assume that the former would be more probable than the latter, based on conditional probability counts of the true. This is a hypothesis we test in our model, and it indeed turns out to be true. A snapshot of our test file is in Fig 5. Overall, while this model performs well, there remain controversies as to how well such models relate to human cognitive processes.

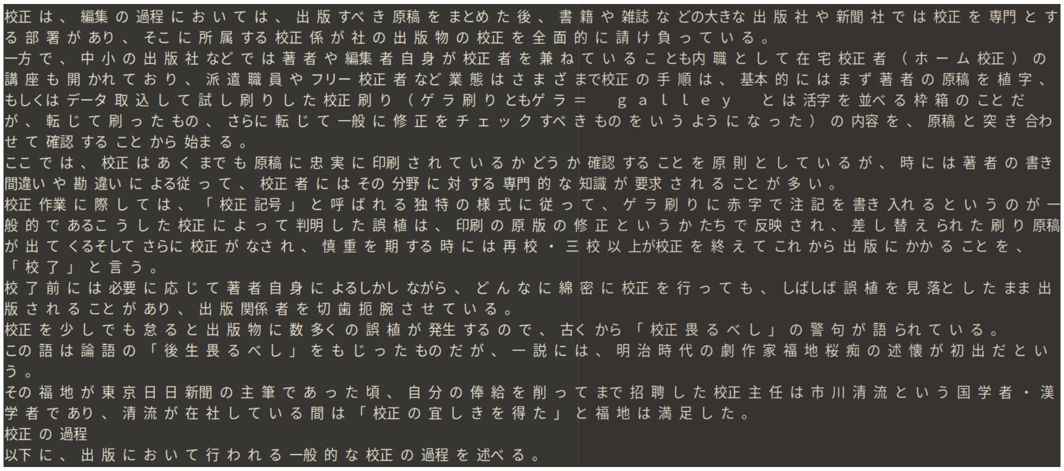
4.4 Extra: Testing on Japanese

While English employs word spacing, which makes word segmentation from written corpora fairly easy, Japanese does not (and neither do Mandarin, Cantonese and agglutinative languages). This makes Japanese word segmentation a very important problem that every language model based in Japanese needs to face. We trained a bigram model on Wikipedia Japanese data and tested it on Zhang Lang’s corpus to come up with the following word segmentation, a snapshot of which is in Fig 6.



```
>>> segment('helloiam')
['hello', 'i', 'am']
>>> segment('howareyoudoing')
['how', 'are', 'you', 'doing']
>>> segment('iamworkingonaproject')
['i', 'am', 'working', 'on', 'a', 'project']
>>> segment('onceuponatime')
['once', 'upon', 'a', 'time']
>>> segment('choosespain')
['choose', 'spain']
```

Figure 5: Test Set Snapshot



```
校正 は 、 編集 の 過程 に お い て は 、 出 版 すべ き 原稿 を まとめ た 後 、 書 籍 や 雑誌 な どの大きな 出 版 社 や 新聞 社 で は 校正 を 専門 と す
る 部 署 が あり 、 そこ に 所 属 する 校正 係 が 社 の 出 版 物 の 校正 を 全 面 的 に 請 け 負 っ て い る 。
一方 で 、 中 小 の 出 版 社 など で は 著 者 や 編集 者 自 身 が 校正 者 を 兼 ね て い る こ とも内 職 と し て 在 宅 校正 者 （ ホ ー ム 校正 ） の
講 座 も 開 かれ て お り 、 派 遣 職 員 や フリー 校正 者 など 業 態 は さ ま ざ まで校正 の 手 順 は 、 基本 的 に は ま ず 著 者 の 原稿 を 植 字 、
もしくは データ 取 込 し て 試 し 刷 り し た 校正 刷 り （ ゲ ラ 刷 り ともゲ ラ ＝  ｇ ａ ｌ ｌ ｅ ｙ  と は 活字 を 並べ る 枠 箱 の こと だ
が 、 転 じ て 刷 っ た もの 、 さらに 転 じ て 一般 に 修 正 を チ ェ ッ ク すべ き もの を い う よう に な っ た ） の 内容 を 、 原稿 と 突 き 合わ
せ て 確認 する こと から 始ま る 。
ここ で は 、 校正 は あ く まで も 原稿 に 忠 実 に 印刷 さ れ て い る か どう か 確認 する こと を 原 則 と し て い る が 、 時 に は 著 者 の 書き
間違い や 勘 違い に よる従 っ て 、 校正 者 に は その 分野 に 対 する 専門 的 な 知識 が 要求 さ れ る こと が 多 い 。
校正 作業 に 際 し て は 、 「 校正 記号 」 と 呼 ば れ る 独 特 の 様 式 に 従 っ て 、 ゲ ラ 刷 り に 赤 字 で 注 記 を 書き 入れ る と い う の が 一
般 的 で あるこ う し た 校正 に よ っ て 判明 し た 誤 植 は 、 印刷 の 原 版 の 修 正 と い う か たち で 反映 さ れ 、 差 し 替 え られ た 刷 り 原稿
が 出 て くるそして さらに 校正 が なさ れ 、 慎 重 を 期 する 時 に は 再 校 ・ 三 校 以 上が校正 を 終 え て これ から 出 版 に かか る こと を 、
「 校 了 」 と 言 う 。
校 了 前 に は 必要 に 応 じ て 著 者 自 身 に よるしかし ながら 、 ど ん な に 綿 密 に 校正 を 行 っ て も 、 しばしば 誤 植 を 見 落と し た まま 出
版 さ れ る こと が あり 、 出 版 関係 者 を 切 歯 扼 腕 さ せ て い る 。
校正 を 少 し で も 怠 る と 出 版 物 に 数 多く の 誤 植 が 発生 する の で 、 古く から 「 校正 畏 る べ し 」 の 警 句 が 語 られ て い る 。
この 語 は 論 語 の 「 後 生 畏 る べ し 」 を も じ っ た もの だ が 、 一 説 に は 、 明 治 時 代 の 劇 作 家 福 地 桜 痴 の 述 懐 が 初 出 だ と い
う 。
その 福 地 が 東 京 日 日 新聞 の 主 筆 で あ っ た 頃 、 自 分 の 俸 給 を 削 っ て まで 招 聘 し た 校正 主 任 は 市 川 清 流 と い う 国 学 者 ・ 漢
学 者 で あり 、 清 流 が 在 社 し て い る 間 は 「 校正 の 宜 し き を 得 た 」 と 福 地 は 満 足 し た 。
校正 の 過程
以下 に 、 出 版 に お い て 行 わ れ る 一般 的 な 校正 の 過程 を 述べ る 。
```

Figure 6: Japanese Word Segmentation

5 A Statistical Approach

We turn our focus to Venkataraman [9]'s model, which relies on the probability that a word w_i appears given that some previous words $w_{i-1}, w_{i-2}, \ldots$ have already appeared. We estimate the necessary n-grams probabilities in function of other n-grams of lower order and, when we get to 1-grams, we back off to the relative frequencies of the phonemes of a given word. The model only focuses on 1-grams, 2-grams, and 3-grams, though this could be extended if necessary. Fig 7 shows a complete description of how we calculate these probabilities. Unlike the focus of the previous models we have seen so far, we will focus on the **algorithmic level** description of this model.

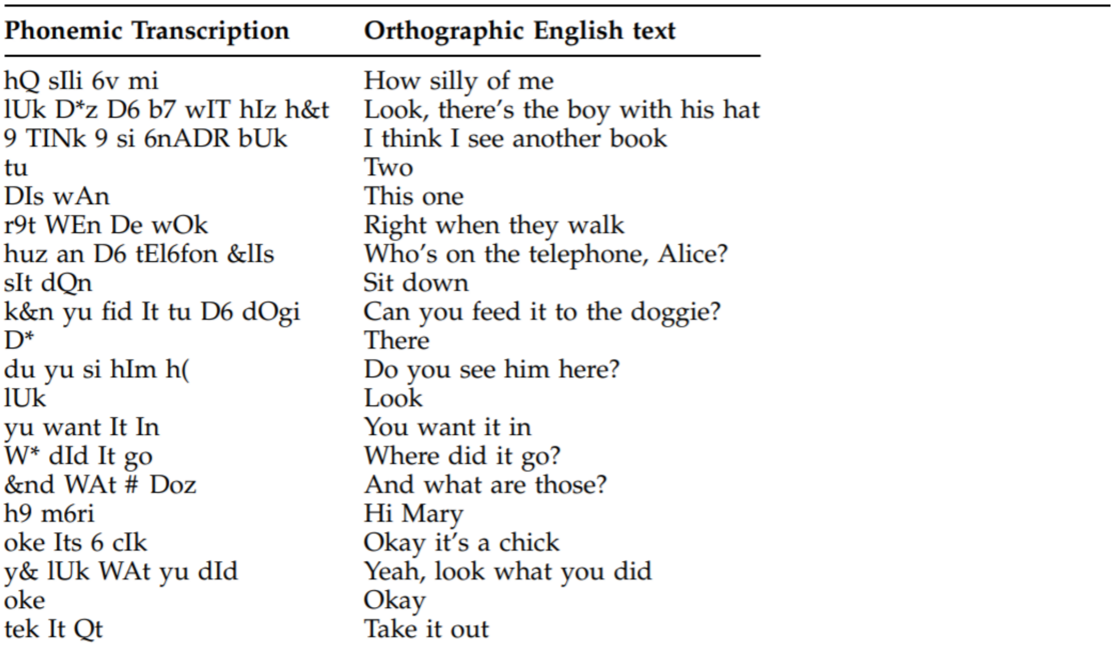
5.1 Dataset

We use the same dataset as Brent [1], that consists of transcripts made by Bernstein-Ratner (1987) of the CHILDES Database (MacWhinney and Snow 1985). This dataset consists of nine mothers talking freely to their children (13-21 months old),

$$
\begin{aligned}
\mathrm{P}(w_i \mid w_{i-2}, w_{i-1}) &= \begin{cases} \frac{S_3}{N_3+S_3}\frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-1},w_i)} & \text{if } C(w_{i-2}, w_{i-1}, w_i) > 0 \\ \frac{N_3}{N_3+S_3}\mathrm{P}(w_i \mid w_{i-1}) & \text{otherwise} \end{cases} \\
\mathrm{P}(w_i \mid w_{i-1}) &= \begin{cases} \frac{S_2}{N_2+S_2}\frac{C(w_{i-1},w_i)}{C(w_i)} & \text{if } C(w_{i-1}, w_i) > 0 \\ \frac{N_2}{N_2+S_2}\mathrm{P}(w_i) & \text{otherwise} \end{cases} \\
\mathrm{P}(w_i) &= \begin{cases} \frac{C(w_i)}{N_1+S_1} & \text{if } C(w_i) > 0 \\ \frac{N_1}{N_1+S_1}\mathrm{P}_\Sigma(w_i) & \text{otherwise} \end{cases} \\
\mathrm{P}_\Sigma(w_i) &= \frac{r(\#) \prod_{j=1}^{k_i} r(w_i[j])}{1 - r(\#)}
\end{aligned}
$$

Figure 7: N_i denotes the number of distincts i-grams, S_i is the some of their frequencies, C() is the count function, r() denotes the relative frequency function. Taken from Venkataraman [9]

which we hope will give us a good estimate on how children understand a speech stream. Fig 8 shows some examples of this dataset.

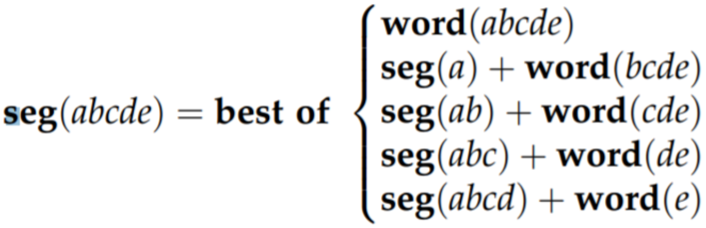


Phonemic Transcription	Orthographic English text
hQ sIli 6v mi	How silly of me
lUk D*z D6 b7 wIT hIz h&t	Look, there's the boy with his hat
9 TINk 9 si 6nADR bUk	I think I see another book
tu	Two
DIs wAn	This one
r9t WEn De wOk	Right when they walk
huz an D6 tEl6fon &lIs	Who's on the telephone, Alice?
sIt dQn	Sit down
k&n yu fid It tu D6 dOgi	Can you feed it to the doggie?
D*	There
du yu si hIm h(	Do you see him here?
lUk	Look
yu want It In	You want it in
W* dId It go	Where did it go?
&nd WAt # Doz	And what are those?
h9 m6ri	Hi Mary
oke Its 6 cIk	Okay it's a chick
y& lUk WAt yu dId	Yeah, look what you did
oke	Okay
tek It Qt	Take it out

Figure 8: Twenty randomly chosen examples from the input corpus, written with their orthographic transcripts. Taken from Venkataraman [9]

5.2 Algorithm

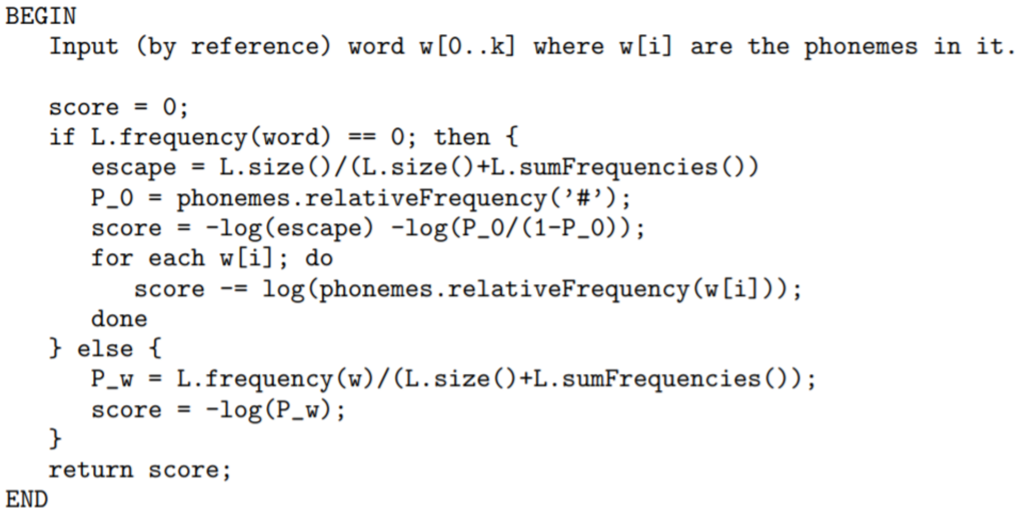
To find word boundaries in a given utterance, we try to split it at a given place and check what the *score* is. Then we take the lowest score of all the segmentations. For example, Fig 9 represents what we would do if we were trying to segment the word $s = abcde$



$$\mathbf{seg}(abcde) = \mathbf{best\ of} \begin{cases} \mathbf{word}(abcde) \\ \mathbf{seg}(a) + \mathbf{word}(bcde) \\ \mathbf{seg}(ab) + \mathbf{word}(cde) \\ \mathbf{seg}(abc) + \mathbf{word}(de) \\ \mathbf{seg}(abcd) + \mathbf{word}(e) \end{cases}$$

Figure 9: Example to segment s = abcde. Taken from Venkataraman [9]

where **seg** is our function of interest and **word** is a way of scoring each word we find. A pseudocode description of that function is shown in Fig 10.



```
BEGIN
   Input (by reference) word w[0..k] where w[i] are the phonemes in it.

   score = 0;
   if L.frequency(word) == 0; then {
      escape = L.size()/(L.size()+L.sumFrequencies())
      P_0 = phonemes.relativeFrequency('#');
      score = -log(escape) -log(P_0/(1-P_0));
      for each w[i]; do
         score -= log(phonemes.relativeFrequency(w[i]));
      done
   } else {
      P_w = L.frequency(w)/(L.size()+L.sumFrequencies());
      score = -log(P_w);
   }
   return score;
END
```

Figure 10: Description of `evalWord`. If the word is novel, then the model uses a distribution over the phonemes of the word. Taken from Venkataraman [9]

5.3 Results

In order to avoid any bias the original corpus might have in terms of the order the sentences are presented, we first shuffle all the sentences before processing. Below are shown some examples of the segmentations that this model was able to perform.

```
"""
For #6n El6f6nt#, the segmentation is 6n El#6f6nt#
For #pUl It Qt kwIk#, the segmentation is pU#l #It #QtkwIk#

For #WAts DIs gel#, the segmentation is WAt#s DI#s gel#

For #WAts D&t#, the segmentation is WAt#s D&t#

For #WAts DIs#, the segmentation is WAt#s DIs#
"""
```

Even though the algorithm used for this is fairly simple, we can still see a reasonable segmentation for those utterances and thus conclude the usefulness of this method.

5.4 Model improvement

In order to obtain a more accurate segmenter, we could try using a larger input corpus (currently it onlye contains 9790 utterances and 33,399 words) and have longer utterances to update our model with more significant data.

On the other hand, we could change the way that **evalWord** works when finding novel words. Currently, it focuses on the statistical frequency of the phonemes of the word. We could, for example, have this value be drawn from another probability distribution instead. In Lecture 3, we are told that *A representation of degrees of belief in terms of probability theory is necessary to cohere with common sense*, which is exactly what we are trying to achieve. As such, we believe that, for example, putting prior probabilities on the types of words our model tries to learn will significantly improve its performance. We could use this technique to prevent the model from picking segmentations that are not phonotactically correct. This is similar in nature to the discussion in class around priors in The Number Game, where we assigned low priors to hypotheses that go against “common sense".

6 Conclusion

In this paper, we have detailed how the computational model based on Saffran et al. [7]'s infant learning experiment, as well as those based on Segaran and Hammerbacher [8], and Venkataraman [9] function, how they are implemented, and how they perform on the segmentation task given varying kinds of inputs—whether it be nonce words, English, or Japanese. We have also analysed how these models relate with the human intuition and cognitive experiments. Finally, we discussed some of the advantages and disadvantages of these models, especially in relation to human cognition, as well as ways to improve on the models and directions for future work.

7 Future Work

We wish to extend this work in the future, both along the experimental cognitive science and the computational directions. Our future plans include

- collecting our own human data for nonce speech streams (similar to Saffran et al.'s) and seeing how our PCFG model compares to human learners
- simulating a beginner language learner with PCFGs—one who adjusts the probability in their mental PCFG representation for every new syllable seen, and then investigating into how the rules change (this is something one of us is working on from a phoneme-learning point-of-view on another project and finds really interesting)
- extending the probabilistic `n`-gram word segmentation algorithm to other languages

8 Contribution

Shinjini has developed, implemented and tested the PCFG model as well as implemented and tested the first probabilistic `n`-gram model mentioned. She has also written out the corresponding sections in the paper, as well as the Abstract, Motivation and Future Work sections.

Raul has helped in testing of the PCFG model, and has implemented and tested the second statistical model mentioned. He has written out the corresponding sections in the paper, as well as the Introduction and the Conclusion.

Bibliography

[1] M. Brent. An Efficient, Probabilistically Sound Algorithm for Segmentation and Word Discovery. *Machine Learning*, 34, 1999. doi: 10.1023/A:1007541817488.

[2] S. Goldwater, T. L. Griffiths, and M. Johnson. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21 – 54, 2009. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2009.03.008. URL `http://www.sciencedirect.com/science/article/pii/S0010027709000675`.

[3] N. D. Goodman, J. Tenenbaum, and T. Gerstenberg. Concepts in a Probabilistic Language of Thought. 2014.

[4] M. Johnson, T. Griffiths, and S. Goldwater. Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *HLT-NAACL*, 2007.

[5] S. T. Piantadosi, J. B. Tenenbaum, and N. D. Goodman. Bootstrapping in a language of thought: a formal model of numerical concept learning. *Cognition*, 123(2):199–217, 2012. doi: doi/10.1016/j.cognition.2011.11.005.

[6] S. T. Piantadosi, J. B. Tenenbaum, and N. D. Goodman. The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, 123(4):392–424, 2016. doi: doi/10.1037/a0039980. URL `https://psycnet.apa.org/doi/10.1037/a0039980`.

[7] J. R. Saffran, R. N. Aslin, and E. L. Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926 – 1928, 1996. doi: http://linguistics.berkeley.edu/~kjohnson/ling290e/saffran_et_al_1996.pdf. URL `http://www.sciencedirect.com/science/article/pii/S0010027709000675`.

[8] T. Segaran and J. Hammerbacher. *Beautiful Data:*. OReilly, 2009.

[9] A. Venkataraman. A Statistical Model for word Discovery in Transcribed Speech. *Comput. Linguist.*, 27(3):352–372, Sept. 2001. ISSN 0891-2017.