
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
ralcant@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. [8]). 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 [10]) and Bayesian realms (Goldwater et al. [3]). 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.

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 rely on segmenting words correctly. We wish to distill the knowledge gained from our understanding of human cognition into computational models and human-like intelligent systems.

3 Saffran Revisited, Computationally

Saffran et al. [8]’s groundbreaking paper delves into statistical learning by 8-month-old infants, and aims to probe the 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, suggest 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. [4], Piantadosi et al. [7], Piantadosi et al. [6] 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. [3] approaches 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. [5], 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 4-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.

```

1      """
2      Sentence -> Words [1.0]
3      Words -> Word Words [0.8] | Word [0.2]
4      Word -> Syllables [1.0]
5      Syllables -> Syllable Syllables [0.8] | Syllable [0.2]
6      Syllable -> 'tu' [0.083]
7      Syllable -> 'pi' [0.168]
8      Syllable -> 'ro' [0.083]
9      Syllable -> 'go' [0.083]
10     Syllable -> 'la' [0.168]
11     Syllable -> 'bu' [0.083]
12     Syllable -> 'da' [0.083]
13     Syllable -> 'ko' [0.083]
14     Syllable -> 'ti' [0.083]
15     Syllable -> 'du' [0.083]
16     """

```

Just like Saffran et al., we generate speech stream by randomly concatenating words from the input vocabulary. 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.

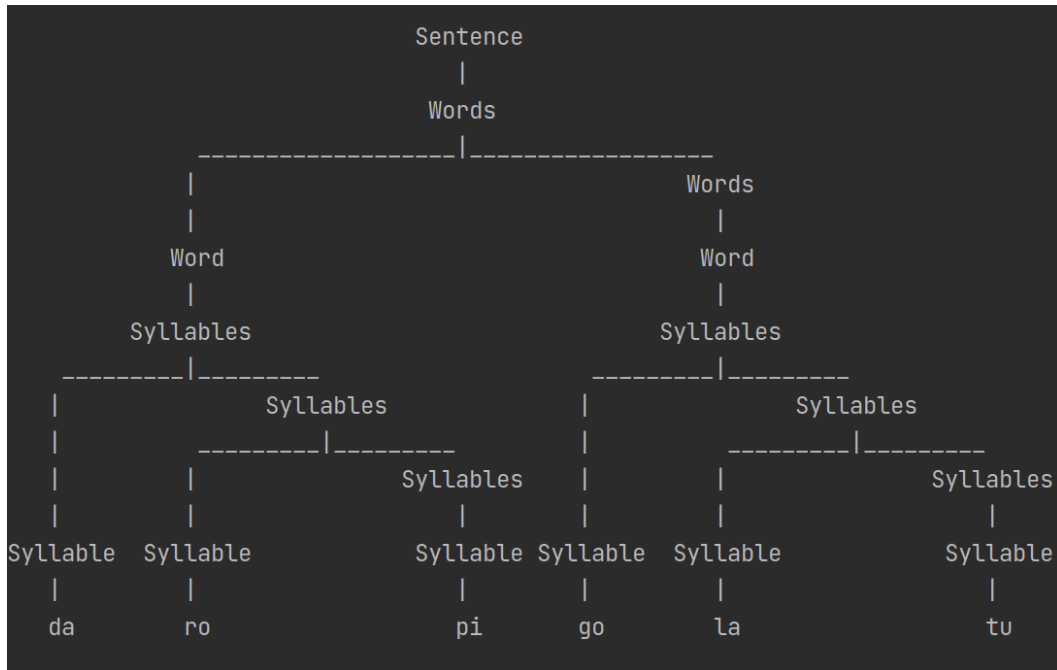


Figure 1: Sample Parse Tree with High Probability

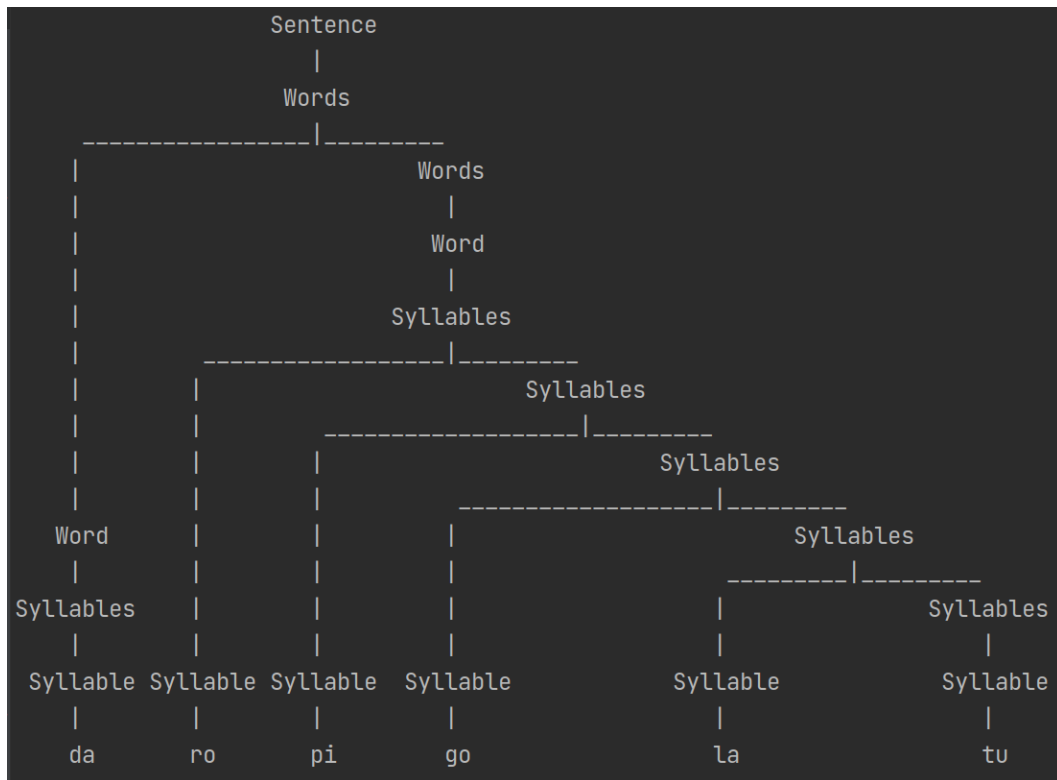


Figure 2: Sample Parse Tree with Low Probability

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 pretty 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 [9].

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, unknified, 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 pretty 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ある。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＝galley 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を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りとりながら、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、出版関係者を切齒扼腕させている。

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

Unlike the PCFG approach, Venkataraman [10]’s model relies more on the relative frequency of certain words and on the probability that a word w_i appears given that some previous words w_1, w_2, \dots, w_{i-1} have already appeared. It computes the most likely segmentation of a sentence (or *utterance*) using a dynamic programming search method, known as the viterbi algorithm (Forney [2]). For this work, we focus on phonemes’ frequencies in each utterance and also in i -grams to take into consideration the relative frequency that sequences of words appearing in our corpus. Venkataraman’s model focuses on $i = 1, 2, 3$.

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),

which we hope will give us a good estimate on how children understand a speech stream.

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 7 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 7: Example to segment $s = abcde$. Taken from Venkataraman [10]

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 8.

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

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 DI#s gel#, the segmentation is WAt#s DI#s gel#

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

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

"""

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

6 Conclusion

In this paper, we have detailed how the computational model based on Saffran et al. [8]’s infant learning experiment, as well as those based on Segaran and Hammerbacher [9], and Venkataraman [10] 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] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973. doi: 10.1109/PROC.1973.9030.
- [3] 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>.
- [4] N. D. Goodman, J. Tenenbaum, and T. Gerstenberg. Concepts in a Probabilistic Language of Thought. 2014.
- [5] M. Johnson, T. Griffiths, and S. Goldwater. Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *HLT-NAACL*, 2007.
- [6] 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](https://doi.org/10.1016/j.cognition.2011.11.005).
- [7] 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](https://doi.org/10.1037/a0039980). URL <https://psycnet.apa.org/doi/10.1037/a0039980>.
- [8] 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>.
- [9] T. Segaran and J. Hammerbacher. *Beautiful Data*:. OReilly, 2009.
- [10] A. Venkataraman. A Statistical Model for word Discovery in Transcribed Speech. *Comput. Linguist.*, 27(3):352–372, Sept. 2001. ISSN 0891-2017.