Word segmentation and lexicon learning from child-directed speech using multiple cues

Zébulon Y. Goriely
Queens' College

UNIVERSITY OF
CAMBRIDGE

A dissertation submitted to the University of Cambridge
in partial fulfilment of the requirements for being
a candidate for Part III of the Computer Science Tripos

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
UNITED KINGDOM

Email: zg258@cam.ac.uk

10/06/2021

Declaration

I Zébulon Y. Goriely of Queens' College, being a candidate for Computer Science Tripos, Part III, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 11,982[1]

Signed: Zébulon Y. Goriely

Date: 10/06/2021

Acknowledgements

Thanks to Paula Buttery and Andrew Caines for supervising me, guiding me through my project and for the great side-tracked discussions about language. Thank you to Alastair Beresford for being my Director of Studies throughout the last four years and for providing excellent feedback on my work. Thank you to my friends, for helping me through this troubled year, and to my loving parents, for their continuous and constant support.

[1] `texcount dissertation.tex -sub=chapter`, adding the words in text, words in headers and words outside text.

Abstract

Word segmentation and lexicon learning from child-directed speech using multiple cues

One of the first tasks children undergo when acquiring language is word segmentation, identifying word boundaries in a continuous speech stream. The *word segmentation problem* refers to how children are able to learn to segment speech, given that there is no clear demarcation of words. Differing views of speech processing consider whether lexical recognition is used to bootstrap this process, or whether children make use of sub-lexical properties such as prosody and predictability statistics. This has prompted the creation of a variety of computational models for child speech segmentation, to explain how the word segmentation problem is solved and what linguistic properties are used to solve it.

In this study, I examine two state-of-the-art frameworks for child speech segmentation, PHOCUS and MULTICUE. PHOCUS is driven by lexical recognition, whereas MULTICUE combines sub-lexical properties to make boundary decisions. I replicate these frameworks, confirm their findings and develop two major improvements to MULTICUE that significantly improve its performance. I also perform novel benchmarking of these two frameworks, comparing them across a variety of experiments.

Taking inspiration from both frameworks, I design a new framework for segmentation: the DYnamic Programming MULTIple-cue framework, or DYMULTI. This framework combines the strengths of PHOCUS and MULTICUE and is able to consider both sub-lexical and lexical cues when making boundary decisions. It does so while learning incrementally and unsupervised, beginning with no language-specific knowledge.

I achieve state-of-the-art results when running DYMULTI on the standard corpus used for evaluation, outperforming prior work. I also perform cross-lingual evaluation, finding that DYMULTI outperforms PHOCUS and MULTICUE on 15 of 26 different languages. As a model built on psycholinguistic understanding, this validates DYMULTI as a robust model for speech segmentation and a contribution to the understanding of language acquisition, with impacts ranging from speech-to-text systems to the teaching of language itself.

Total word count: 11,982[2]

[2] `texcount dissertation.tex -sub=chapter`, adding the words in text, words in headers and words outside text.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

Unlike many written languages, where words are separated by spaces, spoken communication is delivered in continuous utterances with only occasional pauses and no clear demarcation of words (Cole and Jakimik, 1980). Yet, adults are usually able to segment speech with no problem, without even realising that there are no such markings. They are assisted in part by fully developed lexicons, which they use to identify familiar words in the speech stream. Children, on the other hand, are born with no lexicon to consult, yet by the age of six months they are already capable of segmenting the speech stream into words and phrasal units (Bortfeld et al., 2005; Thiessen and Saffran, 2007; Jusczyk, 1999).

The question of how children are able to learn to segment speech and bootstrap their lexicons is the **word segmentation problem**. In the 1980s and 1990s there was a renewed interest in examining the statistical properties of language, and in particular how these may impact the understanding of language acquisition and comprehension (Christiansen et al., 1998). Psycholinguistic studies from this time found that children use statistical properties of language to help solve the word segmentation problem. Such properties include lexical stress (Cutler and Carter, 1987; Cutler and Mehler, 1993; Jusczyk, 1999), phonotactics (Jusczyk et al., 1993; Mattys et al., 1999; Mattys and Jusczyk, 2001), predictability statistics (Saffran et al., 1996a,c; Thiessen and Saffran, 2003), allophonic differences (Jusczyk et al., 1999a), coarticulation (Johnson and Jusczyk, 2001), vowel harmony (Suomi et al., 1997) and prosody (Cooper and Paccia-Cooper, 1980; Gleitman et al., 1988).

Interest in the segmentation problem, combined with the evidence provided by these psycholinguistic studies, has led to the design of a variety of computational models for an abstract version of the task. In the established paradigm, utterances are represented symbolically as strings of phones or phonemes without word boundaries, and models have the task of finding these boundaries without supervision. Besides offering insight into the segmentation problem, such models have also developed into successful algorithms for segmenting written text in languages where word boundaries are not marked (Feng et al., 2004; Sproat and Shih, 1990).

In this study, I compare two approaches to such cognitive models; the *boundary-finding* approach

and the *language modelling* approach. The boundary-finding approach considers statistical information present around each inter-phoneme position to make local boundary decisions, often operating phone-by-phone. Language modelling methods operate utterance-by-utterance, calculating the most-likely segmentation of each, based on lexical recognition. I re-implement the top-performing models for these two approaches, PHOCUS (Venkataraman, 2001; Blanchard et al., 2010) and MULTICUE (Çöltekin and Nerbonne, 2014; Çöltekin, 2017), both of which achieve similar scores on the BR corpus, the de-facto standard for evaluating segmentation models (Brent, 1999).

Through the comparison of these two approaches, I observe that the boundary-finding methods can combine information from multiple sub-lexical cues, but cannot make decisions based on the placement of other boundaries. I also find that the language modelling methods can make decisions based on the placement of other boundaries, but cannot combine information from multiple sub-lexical cues.

Based on these observations, I develop the DYnamic programming MULTIple-cue (DYMULTI) framework for modelling word segmentation. This framework combines the strengths of both approaches and allows for the consideration of sub-lexical and lexical cues, achieving higher F_1-scores scores on the BR corpus than any previous state-of-the-art model that uses the same constraints. I also undertake novel cross-lingual evaluation of these models, finding that my model outperforms PHOCUS and MULTICUE on 15 of 26 languages, confirming its validity as a cognitive model for infant word segmentation.

1.1 Contributions

The contributions of this project are as follows:

- A detailed description of the word segmentation problem and the previous psycholinguistic and computational modelling studies that have investigated it (chapter 2).
- An re-implementation of several state-of-the-art PHOCUS and MULTICUE models (sections 3.1 and 3.2).
- A new set of cues for MULTICUE and a novel unsupervised weighted majority-vote algorithm, both of which significantly improve its performance, outperforming prior work (sections 3.1.1, 3.1.2, 5.1.1 and 5.1.2).
- A novel framework for speech segmentation, DYMULTI, which outperforms previous state-of-the-art models, achieving the highest F_1-scores to date (section 3.3 and chapter 5).
- A more thorough and robust benchmarking of state-of-the-art segmentation models than has previously been performed, comparing the PHOCUS, MULTICUE and DYMULTI frameworks. This includes cross-lingual evaluation across 26 languages, an investigation into the effect of utterance-order, and a comparison of learning rates across models (chapter 5).

As the majority of previous studies do not test their models on more than two languages, the cross-lingual evaluation is a particularly noteworthy contribution, and the state-of-the-art performance of DYMULTI across these languages marks a significant advancement to the field, furthering our understanding of how children acquire language. These findings have potential impacts on how language is taught, our understanding of human speech processing, application of that understanding for speech recognition technology, and even for remedial work with language disorders such as delay, aphasia or dyslexia.

As part of this study, I also released my implementations of PHOCUS, MULTICUE and DYMULTI as an open-sourced repository on Github, consisting of 4000 lines of Python code. The `wordseg` library was used for its command-line interface, for pre-processing corpora and evaluation (Bernard, 2018). The re-implementation of PHOCUS and MULTICUE was completed without access to the original source code. No work presented in this dissertation was undertaken before the start of Part III.

Chapter 2

Background

In this chapter, I give the psycholinguistic background to the word segmentation problem. I then discuss the computational models that have been designed to explore it, detailing the boundary-finding and language-modelling approaches.

2.1 Cues for segmentation

Despite the lack of consistent acoustic gaps between spoken words, adults are able to segment the speech stream into linguistically significant units and therefore access their meaning, a process called *segmentation*. Early models of speech processing declared segmentation to be a by-product of lexical identification (Cole and Jakimik, 1978; Marslen-Wilson and Welsh, 1978), later described as 'serendipitous' or 'interactionist' segmentation models (Cutler, 1996; Cairns et al., 1997). These models identify words in the speech stream by matching them against the listener's lexicon, either processing the utterance in a strictly temporal order (Marslen-Wilson and Welsh, 1978) or by using the activation of competing lexical items to cut up the input (McClelland and Elman, 1986). These models can make use of sub-lexical cues, such as adults' sensitivity to phonotactic information, to make judgements about 'possible' words (Greenberg and Jenkins, 1966), but are fundamentally driven by the lexicon.

Another view of speech processing is that segmentation occurs purely on the basis of information in the speech signal without making use of any lexical influences. Cutler (1996) call these 'explicit' segmentation models, and multiple studies have found that adults can segment purely using low-level information. Saffran et al. (1996c), for example, found that within 20 minutes of exposure to an artificial language, adults are able to use phonotactic information to tell non-words apart from words in a speech stream. Such studies do not refute the interactionist accounts, as these can still incorporate low-level information, but they do show that adult segmentation is not fully driven by lexical recognition.

When it comes to infants, there is evidence that lexical recognition is used to solve the segmentation problem, supporting the interactionist view. Six-month-olds learn new words from

utterances containing familiar names (Bortfeld et al., 2005). French eight-month-olds use function words such as *des* and *mes* for segmentation (Shi and Lepage, 2008). It is not clear whether infants make semantic associations at this stage, but it is clear that they are at least able to recall familiar sound patterns and use them weeks later for segmentation (Jusczyk and Hohne, 1997).

The problem with a model of speech segmentation that only considers lexical recognition lies in explaining how these familiar words are acquired in the first place; infants cannot have any innate assumptions about rhythmic and phonological regularities as these vary between languages (Cutler and Carter, 1987). One hypothesis is that these proto-lexicons are initially populated with single words spoken in isolation (Suomi, 1993). Indeed, in English Parentese (the particular register and style used by caregivers when talking to children), about one-tenth of utterances consist of isolated words (Brent and Siskind, 2001). The issue with this hypothesis is that there is no universal heuristic for identifying single-word utterances and many words will never occur in isolation. Brent and Siskind (2001) claim that if entire multisyllabic utterances are initially added to the lexicon, lexical recognition alone could be sufficient for bootstrapping the lexicon. This claim is supported by a more recent study that found that the proto-lexicon of eleven-month-old French-learning infants contains both words and non-words (Ngon et al., 2013).

On the other hand, there is substantial empirical evidence to suggest that infants use a wide variety of *sub-lexical* cues to solve the initial segmentation problem. Many of these are based on the simple principle that predictability within lexical units is high, and predictability between lexical units is low (Harris, 1955). It was not until the influential studies of Saffran et al. (1996a,b,c) that it became clear that infants are able to use this principle for segmentation. Following their study in adults, they found that infants as young as eight months calculate the *transitional conditional probabilities* of adjacent syllables A and B, defined as

$$\mathrm{TP}(A \rightarrow B) = \frac{\Pr(AB)}{\Pr(A)}, \tag{2.1}$$

where $\Pr(AB)$ is estimated probability of the syllable pair (calculated as the relative frequency) and $\Pr(A)$ is the estimated probability of the syllable A, and use these to place word boundaries when the transitional probability is low (Saffran et al., 1996a,b; Aslin et al., 1998).

These probabilities are also gathered at lower levels. At the phoneme-level, for instance, differences in probabilities between within-word and across-word consonant clusters are used to segment novel phrases such as *fang tine*, as the pair of phones [ŋt] does not occur within English words (Mattys and Jusczyk, 2001). At the lowest level, 7.5-month-old infants use their knowledge of allophonic variations to segment utterances, such as the variants of /t/ and /r/ that distinguish *nitrate* and *night rate* (Jusczyk et al., 1999a).

Infants also seem to be sensitive to prosodic cues, those as young as 7.5 months learn to use the predictable strong-weak stress pattern in English (as in 'BAby') for segmentation (Cutler and Mehler, 1993; Jusczyk et al., 1999b, 1993). While statistical cues may precede stress cues

in their use (Thiessen and Saffran, 2003), stress and coarticulation cues are weighed more heavily by infants once adopted (Johnson and Jusczyk, 2001). Stress alone is unlikely to be a universal cue for segmentation, as it is unclear whether all languages even provide reliable prosodic cues (Saffran et al., 1996c). Indeed, it has generally been accepted that no single cue is solely responsible for solving the segmentation problem, and that a complete model for explicit segmentation must consider information from multiple cues (Jusczyk, 1999; Christiansen et al., 1998; Çöltekin and Nerbonne, 2014; Blanchard et al., 2010).

Taking these accounts together, it is unclear whether initial segmentation in infants is purely explicit, or whether a combination of lexical and sub-lexical information is used. There are many overlapping and competing cues in these studies, so it is difficult to justify one view over the other. For example, segmentation around familiar words could be a result of phonotactic regularity rather than lexical recognition. This motivates the development of computational models in order to test hypotheses in isolation and therefore also solve the word segmentation problem. In particular, the DYMULTI framework developed in this study lets us test whether sub-lexical and lexical cues are alternative or complementary explanations for speech segmentation.

2.2 Segmentation models

Cognitive models for studying the segmentation problem are often designed to study one of two questions: *how* statistical information can be used to segment speech and *what* computational problem is being solved. These are often discussed using terminology from Marr's computational theory of vision (Marr, 1982): the first question operates at Marr's *algorithmic level*, focusing on the algorithm, and the second operates at Marr's *computational level*, focusing on the problem being solved.

Algorithmic-level studies are concerned with the implementation of algorithms that incorporate cognitively plausible mechanisms for the segmentation problem. These models propose efficient algorithms that follow three constraints:

1. They must start with no knowledge of the target language.
2. They must learn unsupervised.
3. They must operate incrementally.

The first constraint follows from the fact that all languages have different phonotactic constraints and vocabularies, yet children can learn any of them. The second is established because children are never explicitly given the boundaries between words, so neither should computational models. The third follows from the fact that we process speech as it is heard, not in batch some time later.

Numerous models have been proposed based on these constraints, taking a wide variety of approaches. Two broad categories stand out: *boundary-finding* methods and *language modelling* methods. These are somewhat related to interactionist and explicit views of speech processing,

although top-performing models make use of both lexical and sub-lexical cues. Investigating these two approaches is the focus of this study.

By contrast, computational-level studies are concerned with defining the goal of segmentation and the logic of the strategy used to meet that goal. As the focus of these studies is not the algorithm, the models developed need not meet the three constraints. An example is the probabilistic model of Goldwater et al. (2009), who find that the assumption that words are statistically independent units leads to under-segmentation by an ideal learner. As a computational-level study, their algorithm does not have to be cognitively plausible. It operates in batch over the corpus, using a hierarchical Dirichlet Process bigram model estimated using a Gibbs sampling algorithm. Besides the batch processing violating the third constraint for an algorithmic-level model, it also takes over 2000 times longer than most algorithmic-level models (Fleck, 2008). In this study, I do not explore these computational-level models, although many algorithmic-level models often offer insight at the computational level.

2.2.1 Boundary-finding methods for segmentation

Boundary finding methods for segmentation relate to the explicit view of speech processing, that segmentation is driven by local information at each inter-phoneme position rather than lexical recognition. Models that use these methods follow directly from experimental studies. For example, Saksida et al. (2017) follow the findings of Saffran et al. (1996a), showing that children segment utterances at low points of transitional probability. Their unsupervised algorithm places boundaries between a syllable pair when the transitional probability of the syllable pair is lower than the two neighbouring pairs. Using syllables as the basic unit of segmentation is widely debated (Coltekin, 2011) and also has a very high baseline as the vast majority of child-directed English words are monosyllabic (Gambell and Yang, 2006).

Earlier studies made use of connectionist models for infant segmentation, as was the trend for investigating many cognitive phenomena at the time (Elman, 1990; Christiansen et al., 1998). As cognitively-plausible models for segmentation need to be unsupervised, these models could not be trained to predict word boundaries directly. Instead, they were often trained on an alternative task. Elman (1990) trained a recurrent neural network to predict phonemes, finding that relatively high error in prediction could indicate word boundaries. Cairns et al. (1994) found that 'peaks' in the error score could also consider word boundaries. Finally, Christiansen et al. (1998) developed a recurrent neural network to predict utterance boundaries, phonemes and lexical stress information in an utterance, finding that the prediction of an utterance boundary was a good indicator of a word boundary. This model allowed them to test these different cues together and in isolation, finding that the best performance was achieved when all three cues were combined.

Inspired by this model, Çöltekin and Nerbonne (2014) developed an explicit model for segmentation, arguing that it is difficult to interpret what connectionist models learn. Their model uses statistical information at each inter-phoneme position, as with transitional probability models,

but extends this by introducing a cue-combination method to combine statistical information from multiple sources, also achieving far better performance than the connectionist models. This is the boundary-finding approach that I re-implement in section 3.1.

2.2.2 Language modelling methods for segmentation

Language modelling methods are based on the interactionist view of speech processing, that segmentation and lexical recognition occur serendipitously, driven by lexical knowledge. These models typically build word *n*-gram models and use statistical criteria to define the 'best' segmentation of an utterance, bootstrapping a lexicon that is then used for further segmentation.

Brent (1999) and Venkataraman (2001) both developed probabilistic language models and used dynamic programming to infer the best segmentation. In Venkataraman's model, the probability of a segmented utterance is given as the joint probability of all words in that utterance. The Viterbi algorithm is then used to find the segmentation that maximises an estimate of this probability. Word probabilities are approximated by *n*-grams, with a back-off procedure to lower-order *n*-grams. As the model processes more utterances, these probability estimates are refined and more words are added to the lexicon, further improving the model. These models produced state-of-the-art results unmatched by boundary-finding methods until later work (Fleck, 2008; Çöltekin, 2017; Çöltekin and Nerbonne, 2014).

It is worth noting that none of the successful language-modelling methods rely only on lexical recognition. A model that only matches utterances with previously-seen utterances will fail, as "short, frequently occurring utterances are likely to be segmented within larger word-level chunks resulting in an over-segmentation of words into their segmental phonology" (Monaghan and Christiansen, 2010). For instance, if 'no' has been added to the lexicon, then 'note' could later be segmented as 'no' and 'te', followed by increasingly smaller segmentations. As such, these models often gather sub-lexical statistics or make sub-lexical assumptions to prevent over-segmentation. The PUDDLE model, for example, uses word-initial and word-final phoneme clusters derived from its lexicon to restrict segmentation (Monaghan and Christiansen, 2010). The model of Venkataraman (2001) incorporates phoneme-level statistics to estimate the probability of unseen words. Finally, the model of Blanchard et al. (2010) extends Venkataraman's model with an additional constraint that all segmented words must contain a syllabic sound. I re-implement these last two models in section 3.2.

2.3 Summary

Experimental psycholinguistic studies provide evidence that infants use sub-lexical statistical and speech cues for solving the segmentation problem, supporting the explicit view of speech processing. Other studies find that infants make use of lexical knowledge, supporting the interactionist view. To study the problem in a controlled manner, computational models have been designed to solve an abstract version of the problem, where continuous speech is represented as

a series of symbolic phonemes. These models either explore what problem is being solved or present cognitively plausible algorithms for solving the problem. To be cognitively plausible, these algorithms must segment incrementally, start with no knowledge of the target language and learn unsupervised. Boundary-finding algorithms correspond to the explicit view of speech processing; and language-modelling algorithms correspond to the interactionist view.

Chapter 3

Design and Implementation

In this chapter, I present my re-implementation of the state-of-the-art models for the boundary-finding approach of Çöltekin and Nerbonne (2014) and the language-modelling approach of Venkataraman (2001) and its extension presented by Blanchard et al. (2010). I discuss the benefits and drawbacks of these two approaches and produce a new model that combines their strengths.

3.1 Çöltekin and Nerbonne's multiple-cue boundary-finding model

The model presented by Çöltekin and Nerbonne (2014) iterates through utterances phoneme-by-phoneme, placing boundaries by combining votes from a set of indicators based on a variety of cues. It is explicit in nature, although it does use statistical cues derived from the lexicon. I refer to this model as MULTICUE. In this section, I describe the model and propose a new variant to the weighted majority-vote algorithm used to combine cues.

3.1.1 Cue combination algorithm

The core strength of MULTICUE lies in its cue combination algorithm, which allows for the consideration of an arbitrary number of psychologically-motivated boundary indicators. As no single cue is solely responsible for segmentation, this allows for a more comprehensive model for explicit segmentation.

Each boundary indicator labels every inter-phoneme position as either 'boundary' or 'word internal'. The model then makes a final decision based on a variation of the weighted majority voting algorithm (Littlestone and Warmuth, 1994). In Çöltekin (2017), the following condition for deciding on the 'boundary' label is given:

$$\sum_{i}^{K} w_i 1_i > \frac{K}{2}, \tag{3.1}$$

where K is the number of boundary indicators, w_i is the weight and 1_i gives the boundary

decision for indicator i, equal to 1 for 'boundary' and 0 for 'word-internal'. Unfortunately, this equation does not make sense for all choices of w_i as the weights of the model are independent and so do not necessarily sum to K. For instance, if all K boundary indicators were assigned a weight of 0.5, the model would never be able to place a boundary. This is because even if all indicators voted for a boundary, the weighted majority vote would only equal $\sum_i^K 0.5 = \frac{K}{2}$.

Instead, my implementation places a boundary if the weighted vote for the 'boundary' label is greater than the weighted vote for the 'word-internal' label:

$$\sum_i^K w_i 1_i > \sum_i^K w_i (1 - 1_i). \tag{3.2}$$

By noticing that the two sides sum to $\sum_i^K w_i$, and so the boundary can simply be placed if the normalised weighted vote exceeds 0.5, this equation can also be rewritten as:

$$\frac{\sum_i^K w_i 1_i}{\sum_i^K w_i} > \frac{1}{2}. \tag{3.3}$$

I believe that it is likely that Çöltekin also implemented this equation, as his results could not have been achieved with equation (3.1).

The majority-vote algorithm is a common and effective method for combining multiple classifiers (Narasimhamurthy, 2005). In this case, the *weighted* majority-vote is used so that votes from boundary indicators that make fewer errors have larger weights. As the model must be unsupervised, the ground-truth boundary locations can not be used to update the weights. Instead, an 'error' happens when an individual cue disagrees with the majority-vote. At each inter-phoneme position, the incremental algorithm gathers votes from each indicator i, decides whether the position is a 'boundary' or is 'word-internal', then increments the error count e_i for each indicator that disagreed with this decision. Finally, the weight w_i of each indicator is updated:

$$w_i = 1 - 2\frac{e_i}{N}, \tag{3.4}$$

where N is total number of inter-phoneme positions seen, producing weights in $[-1, 1]$.

Çöltekin and Nerbonne (2014) state that this update rule "sets the weight of a vote that is half the time wrong to zero, eliminating incompetent voters" and that with this model, the success of boundary decisions depends on the precision of individual boundary indicators. In reality, this score is related to the *accuracy* of these indicators. As there are fewer true 'boundary' labels than 'word-internal' labels, an indicator that never places a boundary will achieve a higher accuracy than an indicator that always places a boundary, so setting weights to zero when the accuracy is

0.5 is misleading. In my implementation, I use the following:

$$w_i = 1 - \frac{e_i}{N}. \tag{3.5}$$

These weights are in the range $[0, 1]$ and are exactly equal to the accuracies of each indicator with respect to the final votes.

A new weighted majority-vote algorithm

As the labels are skewed, I developed a new weighted majority-vote algorithm where a pair of weights is stored for each indicator; p_i for the 'boundary' label and q_i for the 'word-internal' label. Intuitively, this means that model can learn to weigh indicators differently depending on what label they propose. The majority-vote algorithm is adapted so that the 'boundary' label is selected when

$$\frac{\sum_i^K p_i 1_i}{\sum_i^K p_i} > \frac{\sum_i^K q_i (1 - 1_i)}{\sum_i^K q_i}. \tag{3.6}$$

The weights for each indicator can then be based on label-specific measures, such as *precision*, *recall* or *F_1-score*. These are standard evaluation measures for computational simulations or classification models where the labels are not equally distributed. In particular, the F_1-score is a better measure of overall performance for each indicator than accuracy. These scores can be calculated for a particular label l as follows:

$$\text{precision}(l) = \frac{c_{l,i}}{T_{l,i}},$$

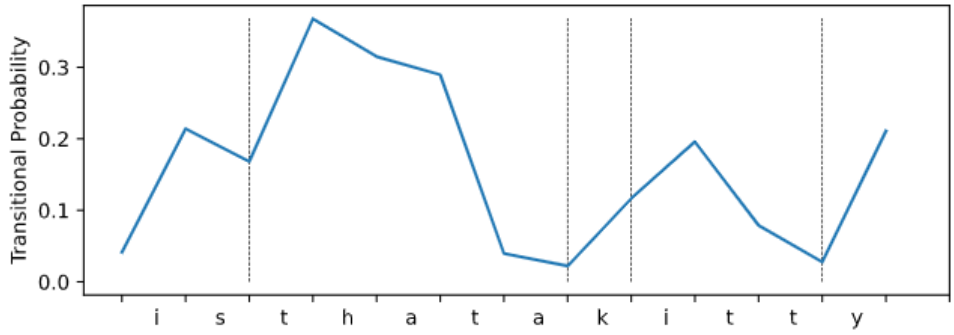
$$\text{recall}(l) = \frac{c_{l,i}}{N_l},$$

$$\text{F}_1\text{-score}(l) = 2 \times \frac{\text{precision}(l) \times \text{recall}(l)}{\text{precision}(l) + \text{recall}(l)},$$

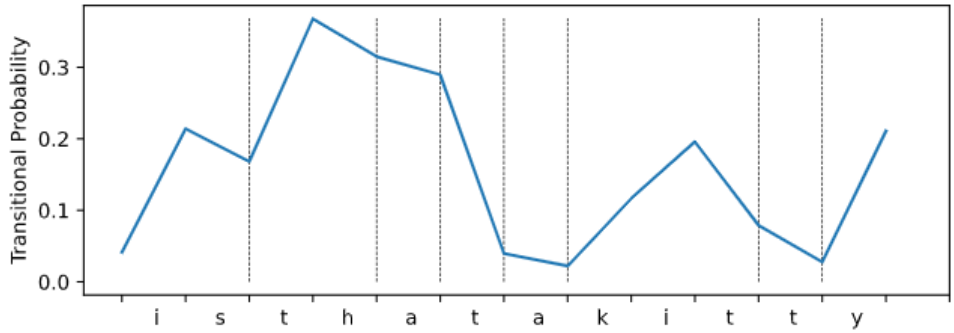
where $c_{l,i}$ is the number of times indicator i correctly suggested label l (TP), $T_{l,i}$ is the number of times label l has been suggested by the indicator (TP + FP) and N_l is the total number of labels l voted for by the final model (TP + FN).

For this model, the two labels are 1 for 'boundary' and 0 for 'word internal' and I examine setting the weights of each indicator to be either the precision, recall or F_1-score for these two labels:

$$p_i = \text{precision}(1) \quad q_i = \text{precision}(0), \tag{3.7}$$



(a) Boundaries before increase in TP.



(b) Boundaries after decrease in TP.

Figure 3.1: Two indicators following the 'partial-peak' strategy, segmenting the phrase 'is that a kitty'. One segments at an increase in transitional probability and the other segments at a decrease. Letters are used instead of phonemes for clarity.

$$p_i = \text{recall}(1) \quad q_i = \text{recall}(0), \tag{3.8}$$

$$p_i = \text{F}_1\text{-score}(1) \quad q_i = \text{F}_1\text{-score}(0). \tag{3.9}$$

The algorithm presented here is the two-label case, but could easily be extended for multiple labels by calculating weights for each label and choosing the label with that gives the highest normalised weighted vote. I evaluate this new majority-vote algorithm against the accuracy-based majority-vote algorithm in section 5.1.2.

3.1.2 Cues and boundary indicators

Using the majority-voting framework, any number of indicators can be considered. Çöltekin and Nerbonne (2014) describes a series of indicators derived from four sets of cues; predictability statistics, utterance boundaries, lexical stress and the lexicon, all deriving from psycholinguistic studies.

All of the indicators calculate a certain measure based on these cues. To propose boundaries, they use a 'partial-peak' strategy. This is based on the 'peak' strategy of transitional probability

models where a boundary would be suggested if the transitional probability at an inter-phoneme position was lower than the transitional probabilities on either side of that boundary. Each cue is split into two indicators, splitting this peak in half. An example is given in figure 3.1, where the first indicator proposes a boundary after a decrease in transitional probability and the other proposes a boundary before an increase in transitional probability. The model can then learn weights associated with each indicator, using the weighted majority-vote algorithm.

Çöltekin and Nerbonne (2014) also include indicators that calculate statistics over a larger context of three phonemes, capturing higher-order regularities, as well as indicators that calculated reverse measures, following the study of Pelucchi et al. (2009) that found that children can also use *reverse* transitional probabilities for segmentation. Çöltekin (2017) later used MULTICUE to explore various predictability cues in isolation. He found that the best performance was achieved when including indicators with a context size of one, two, three and four phonemes and also found that **successor variety** was a better predictability measure than transitional probability.

I reimplemented all cues used by Çöltekin and Nerbonne (2014) as well as the predictability cues used by Çöltekin (2017). Using a simple command-line interface, I re-implemented their models, which I henceforth refer to as MULTICUE-14 and MULTICUE-17 respectively. These models only vary in which indicators are included; MULTICUE-14 has 44 stress, predictability, lexicon and utterance-boundary indicators and MULTICUE-17 has 16 predictability indicators. For my reported results of MULTICUE-17, I use the successor variety predictability cue, as this was the measure that Çöltekin (2017) found to give the best performance. A full description of these cues is provided in appendix A.

An updated set of cues: MULTICUE-21

Based on the success of using a variety of cues (Çöltekin and Nerbonne, 2014) and the success of using higher-order *n*-grams and the successor variety predictability cue (Çöltekin, 2017), I propose a new set of indicators, producing MULTICUE-21. This set consists of the successor variety cue of MULTICUE-17 and the lexicon cues and utterance boundary cues of MULTICUE-14. Indicators are created for *n*-gram values from 1 to 4. The stress cue is not included, following the finding of Çöltekin and Nerbonne (2014) that it decreases performance and also because the cross-lingual corpora that I use for evaluation do not provide stress alignment information.

3.1.3 Model summary

The MULTICUE framework of Çöltekin and Nerbonne (2014) consists of many indicators based on cues known to be used by infants to solve the segmentation problem and combines these cues using a weighted majority-vote algorithm. I presented a new majority-vote algorithm that could be used to set weights according to precision, recall and F_1-score rather than accuracy. I described my implementation of MULTICUE-14 and MULTICUE-17, as well as a new model with a set of cues based on their findings, which I call MULTICUE-21.

3.2 Venkataraman's language-modelling algorithm

Venkataraman's model follows a language-modelling approach to segmentation. As a lexicon is developed and phonemic distributional statistics are learned, utterances are decoded using the Viterbi algorithm to find the maximum-likelihood segmentation. This is an interactionist approach as it is driven by lexical recognition rather than boundary placement.

3.2.1 Language model

A standard language model is used to calculate the likelihood of a segmentation. Given a segmentation $\boldsymbol{W} = w_1, \ldots, w_n$ composed of n individual words $w_i \in \boldsymbol{L}$ in a lexicon $\boldsymbol{L}$, the likelihood of $\boldsymbol{W}$ is

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{W}), \tag{3.10}$$

$$= \underset{\boldsymbol{W}}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i | w_1, \ldots, w_{i-1}). \tag{3.11}$$

To prevent underflow errors in computation, an equivalent calculation is made using log likelihoods:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmin}} \sum_{i=1}^{n} -\log P(w_i | w_1, \ldots, w_{i-1}). \tag{3.12}$$

A common approximation when implementing language models is the n-gram approximation, collapsing these conditional probabilities to consider at most $n-1$ words. Venkataraman (2001) makes a three-gram approximation, estimating $P(w_i | w_{i-2}, w_{i-1})$ with relative frequencies and using a back-off procedure to estimate the probability of unseen n-grams with with lower order n-grams (Katz, 1987). He uses a back-off technique from Witten and Bell (1991), allocating a probability of $\frac{N_n}{N_n + T_n}$ to unseen n-grams. At the lowest stage, unseen word (1-gram) probabilities are estimated by the normalised product of derived phoneme probabilities. The full back-off is given by

$$P(w_i|w_{i-2}, w_{i-1}) = \begin{cases} \frac{T_3}{N_3+T_3} \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-1},w_i)} & \text{if } C(w_{i-2}, w_{i-1}, w_i) > 0 \\ \frac{N_3}{N_3+T_3} P(w_i|w_{i-1}) & \text{otherwise} \end{cases}, \tag{3.13}$$

$$P(w_i|w_{i-1}) = \begin{cases} \frac{T_2}{N_2+T_2} \frac{C(w_{i-1},w_i)}{C(w_i)} & \text{if } C(w_{i-1}, w_i) > 0 \\ \frac{N_2}{N_2+T_2} P(w_i) & \text{otherwise} \end{cases}, \tag{3.14}$$

$$P(w_i) = \begin{cases} \frac{C(w_i)}{N_1+T_1} & \text{if } C(w_i) > 0 \\ \frac{N_1}{N_1+T_1} P_\Sigma(w_i) & \text{otherwise} \end{cases}, \tag{3.15}$$

$$P_\Sigma(w_i) = \frac{r(\#) \prod_{j=1}^{|w_i|} r(w_i[j])}{1 - r(\#)}, \tag{3.16}$$

where N_n is the number of n-gram types and T_n is the number of n-gram tokens. $C(w_i)$ is the frequency of word w_i, '#' is the boundary character, $w_i[j]$ is the jth phoneme of word w_i, $|w_i|$ is the length of w_i in phonemes and $r()$ gives the relative frequency of a phoneme or a boundary.

Venkataraman (2001) implemented 1-gram, 2-gram and 3-gram models, finding a trade-off between precision and recall, with 1-grams giving the best performance overall. This is surprising, as n-gram contexts typically improve performance in such systems. Venkataraman claimed that this is because the 2-gram and 3-gram models are more conservative, as longer n-grams are more infrequent, leading to whole utterances often being inserted into their lexicons. As such, I only implement the 1-gram model (equations (3.15)–(3.16)).

3.2.2 Viterbi search

The language model defines the likelihood of a segmentation, but a search procedure is required to find the most likely segmentation. Exhaustive search is computationally intractable as there are 2^{n-2} possible segmentations for an utterance of n phonemes, so this would be an unreasonable model for human segmentation. Instead, Venkataraman uses Viterbi search (Viterbi, 1967) to decode each utterance, a dynamic programming algorithm that only explores $(n-2)^2$ segmentations.

The algorithm begins with an empty lexicon and no knowledge of phoneme frequencies, building these incrementally as each utterance is processed. The process is unsupervised, as no word boundaries are ever provided to the model. As such, all three constraints for algorithmic-level segmentation are satisfied.

3.2.3 Blanchard's extended algorithm

Blanchard et al. (2010) extend Venkataraman's 1-gram model to produce PHOCUS, for PHonotactic CUe Segmenter. They introduce two phonotactic cues: language-specific and language-universal. The first extends the unseen word estimate to use conditional probabilities of *phoneme* n-grams, rather than the 1-grams of Venkataraman. This cue is language-specific as phonotactic

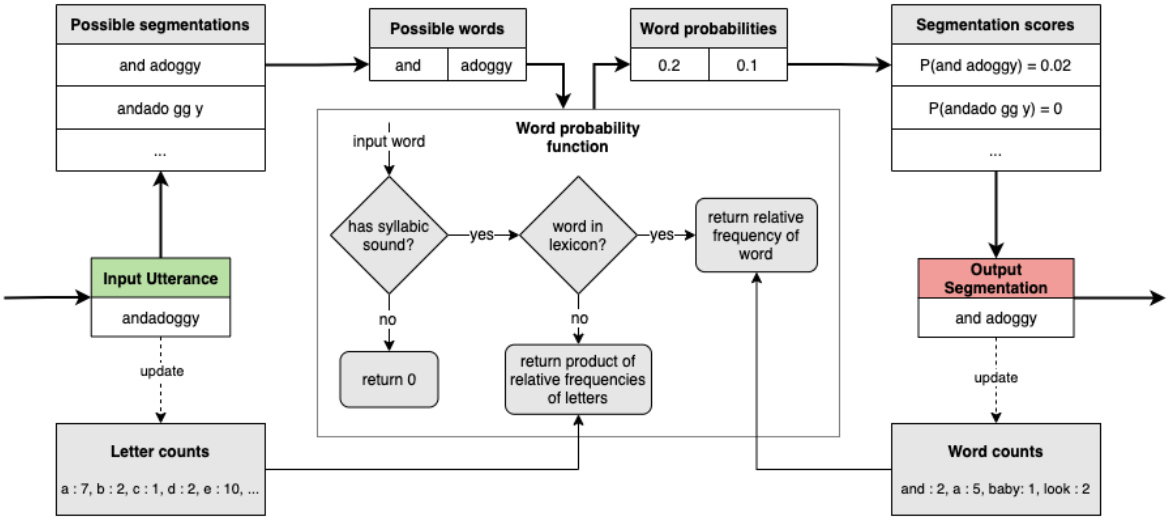


Figure 3.2: A simplified example of PHOCUS-1S segmenting the utterance 'andadoggy', using letters instead of phonemes for clarity.

constraints (permissible phone combinations) vary between languages, so the n-gram probabilities must be learned. The models that keep track of phoneme n-grams are referred to as PHOCUS-n, with PHOCUS-1 being equivalent to Venkataraman's model. For simplicity, I do not consider higher-order n-grams here.

The second cue is the universal constraint that words must have at least one *syllabic* sound. Sounds are considered to be syllabic if they are the nucleus of a syllable in a word. Syllabic sounds in English consist of all vowels and some consonant sounds (such as the [l,m,ɹ, n] sounds in *awful* [ɔfl̩], *rhythm* [ɹɪðm̩], *butter* [bʌtɹ̩] and *even* [ivn̩]). There is much debate about syllables, but they claim that this constraint is plausibly a prior that does not need to be learned, as words contain at least one syllable so must contain at least one syllabic sound. To implement this constraint, probabilities of words that do not have a syllabic sound are set to 0. Adding this constraint to PHOCUS-1 gives PHOCUS-1S.

3.2.4 Full algorithm

PHOCUS-1S iteratively processes each utterance using the Viterbi algorithm to find the segmentation that maximises the product of estimated word probabilities. After segmenting each utterance, the proto-lexicon and phoneme counts are updated, improving the language model. The pseudocode for this algorithm is given in appendix B.1.

A simplified example of this loop is given in figure 3.2, where possible segmentations of the utterance 'andadoggy' are considered. For the first possible segmentation, the probability of the word 'and' is given by its relative frequency in the proto-lexicon, as in equation (3.15). The word 'adoggy' has not been seen before, so its probability is calculated using the relative frequencies of each of its letters, as in equation (3.16). For the second possible segmentation, the word 'gg' contains no syllabic sound, so has a probability of 0, resulting in a probability of 0 for the whole

utterance. Assuming 0.02 is the highest score out of all segmentations considered by the Viterbi algorithm, 'and adoggy' would be selected as the best segmentation for this utterance.

3.2.5 Model summary

The PHOCUS-1 model of Venkataraman (2001), extended by Blanchard et al. (2010) to produce PHOCUS-1S, uses a language model to define the probability of a segmentation based on seen-word frequency and phoneme frequency for unseen words. In PHOCUS-1S, probabilities of words not containing syllabic sounds are set to 0. A Viterbi algorithm finds the most-likely segmentation of an utterance. I implemented both PHOCUS-1 and PHOCUS-1S for comparison with the MULTICUE models.

3.3 Combined segmentation model

MULTICUE is a boundary-finding model for segmentation, but MULTICUE-14 does use indicators based on the lexicon, so it could be considered an interactionist model. PHOCUS is also an interactionist model, using a language model for calculating the probability of segmenting an utterance, but it does use sub-lexical information for estimating the probability of unseen words. Therefore, both models involve a complicated interaction of lexical and sub-lexical information, which is consistent with studies showing that infants use both sources of information for solving the segmentation problem. There are, however, drawbacks to both approaches.

One of the key benefits of MULTICUE is that it can combine an arbitrary number of sub-lexical boundary indicators. This is a good model for explicit segmentation, as experimental studies have shown that infants are sensitive to a wide variety of cues. PHOCUS, on the other hand, cannot consider an arbitrary number of sub-lexical indicators. This is not just a drawback of PHOCUS, but of any language modelling approach to segmentation. To add a new indicator, the entire language model would need to be redefined and this would be very difficult to do without making prior assumptions about the cues.

The strength of PHOCUS lies in the Viterbi search process. The segmentation of an utterance is decided at the lexical level, based on the scores assigned to each word in the segmentation. This means that it is easy to incorporate lexical-level constraints, such as the require-syllabic-sound constraint of Blanchard et al. (2010). Such a constraint can not be easily incorporated into the MULTICUE model, or indeed into any boundary-finding approach to segmentation, as boundary-finding models place boundaries independently of each other using only the local context around that boundary. Hence, the decision cannot depend on the placement of previous or future boundaries.

In this section, I present a new framework segmentation models that combines the two approaches. It collects scores for each inter-phoneme position from multiple indicators using the weighted majority-vote algorithm of MULTICUE, then uses a modification of the Viterbi algorithm from PHOCUS to choose the best segmentation, rather than just placing boundaries

greedily. This combined model allows for the consideration of multiple sub-lexical and lexical cues, addressing the drawbacks of both the boundary-finding and language-modelling approaches to segmentation. I name this framework DYMULTI for DYnamic programming MULTIple-cue model.

3.3.1 Using weighted boundary votes with the Viterbi algorithm

In DYMULTI, the Viterbi algorithm finds the best segmentation according to boundary scores rather than word scores. These boundary scores are adapted from the weighted majority-vote algorithm of the MULTICUE model. Instead of comparing weighted votes for the 'boundary' and 'word-internal' labels, as in equation (3.6), the difference between the two votes is calculated:

$$\text{score}(j) = \frac{\sum_i^K p_i 1_{ij}}{\sum_i^K p_i} - \frac{\sum_i^K q_i (1 - 1_{ij})}{\sum_i^K q_i}, \tag{3.17}$$

where p_i and q_i are the weights for indicator i. These scores lie between -1 and 1 with scores over 0 indicating a boundary and scores close to 1 or -1 suggesting strong agreement between indicators.

Here, p_i and q_i can be the precision, recall, or F_1-score weights as described in section 3.1.1 or can be the accuracy-based weights from the original weighted majority-vote algorithm of MULTICUE by replacing q_i with p_i. The function returns the score at position j in the utterance where 1_{ij} is the vote of indicator i at this inter-phoneme position.

I then adapt the Viterbi algorithm to maximise the sum of these boundary scores, rather than minimise the sum of negative log word probabilities. This adapted algorithm is given in appendix B.2. The word score function now simply returns `score`(j), the score given by the weighted majority-vote algorithm between phoneme $j-1$ and j. At the utterance boundaries, `score`(j) always returns 1.

Without any other changes, this algorithm simply places boundaries at every position where the score is greater than 0, as this maximises the sum over the utterance. As scores over 0 indicate where MULTICUE would have placed a boundary using equation (3.6), this means that DYMULTI will act exactly like MULTICUE if the same indicators are provided. The difference with this new framework is that lexical-level processes can be introduced by adjusting the word score function, as described in the next two sections.

3.3.2 Introducing the require-syllabic-sound lexical constraint

The first lexical-level process I introduce to DYMULTI is the require-syllabic-sound constraint of Blanchard et al. (2010). I adjust the word score function so that if the word has no syllabic sound, the function returns -100. This number is chosen to be far smaller than any positive sum could account for, similarly to the large negative log probability used to simulate a probability of 0 in my implementation of PHOCUS-1S.

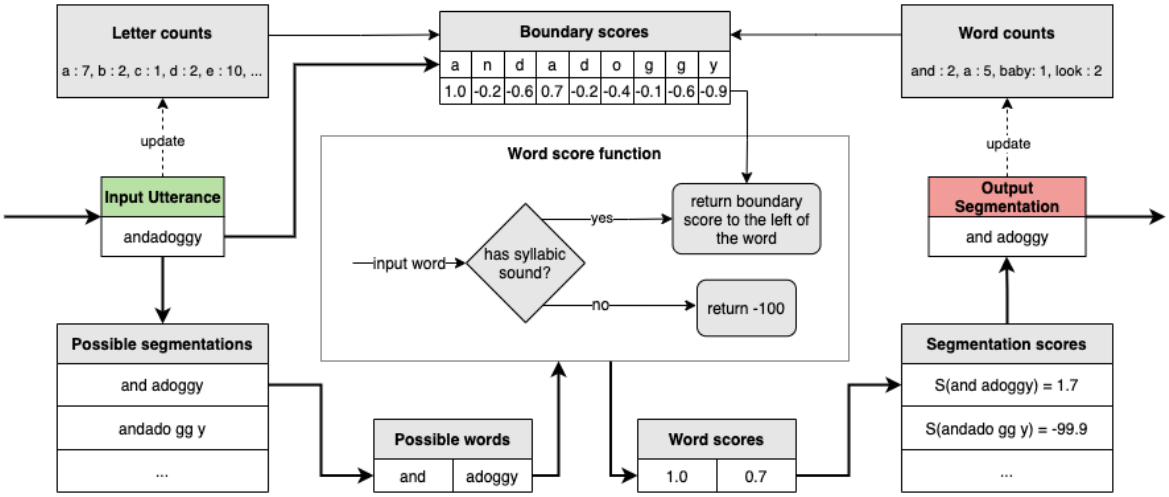


Figure 3.3: A simplified example of DYMULTI segmenting the utterance 'andadoggy', using letters instead of phonemes for clarity.

An example of the DYMULTI framework segmenting the utterance 'andadoggy' is given in figure 3.3. First, the boundary scores for the utterance are calculated using equation (3.17). These scores are then used to calculate the score of each segmentation, using the Viterbi algorithm. For the segmentation 'and adoggy', the boundary scores to the left of the two words are 1.0 and 0.7, so the score for the segmentation is 1.7. As with PHOCUS-1S, the require-syllabic-sound constraint prevents 'andado gg y' from being a valid segmentation, giving a score of -100 to the word 'gg'.

3.3.3 Introducing a lexical recognition model

Using the Viterbi algorithm, other lexical processes can also be introduced to DYMULTI. Here, I propose one such process: a rudimentary lexical recognition process to favour previously-seen words. This mirrors the lexical recognition driving many of the language modelling methods for segmentation (Monaghan and Christiansen, 2010; Venkataraman, 2001; Brent, 1999; Blanchard et al., 2010).

This lexical recognition introduces a single parameter, α, to DYMULTI. To favour previously-seen words, this process simply adds α to the score of a word w if $w \in L$, where L is the proto-lexicon populated with words in previous segmentations. Reasonable values of α lie in $[0, 1]$, where $\alpha = 0$ is equivalent to not using the lexical recognition process. Setting $\alpha = 1$ has the effect of always trying to place boundaries around previously-seen words, as it will always return scores above 0 since $\text{score}(s) \in [-1, 1]$. Note that due to the require-syllabic-sound constraint, these boundaries will not necessarily be placed, but there will still be a very strong bias towards them. Intermediate values of α result in a balance between the lexical recognition process and the boundary-finding process.

The full word score function with both lexical processes is given in figure 3.4 and the pseudocode can be found in appendix B.2.

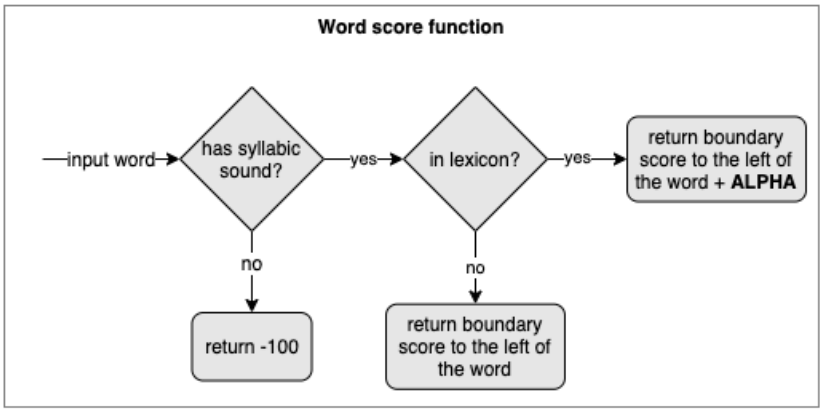


Figure 3.4: The expanded word score function for DYMULTI with the lexical recognition process.

3.3.4 Model summary

The new DYMULTI framework addresses the drawbacks of the language modelling and boundary-finding approaches to segmentation. The model uses the weighted majority-vote algorithm of MULTICUE to produce scores that are then used to select the best segmentation using the Viterbi algorithm of PHOCUS. Using this dynamic programming algorithm, the model is able to incorporate lexical processes. I describe two such processes: the require-syllabic-sound constraint from PHOCUS-1S (Blanchard et al., 2010) and a rudimentary lexical recognition process that takes a single parameter to adjust the weighting given to previously-seen words. The full model efficiently combines multiple sub-lexical and lexical cues for segmentation, without the drawbacks of previous models.

3.4 Summary

In this chapter, I presented my re-implementation of the MULTICUE and PHOCUS models and introduced the new DYMULTI model that combines the strengths of both. I also presented a new majority-vote algorithm for MULTICUE that uses precision, recall and F_1-score to set the weights of each indicator included in the model.

Chapter 4

Data and Evaluation

In this chapter, I discuss the procedure used to evaluate PHOCUS, MULTICUE and DYMULTI. This includes the data used, the baseline segmentation model and the evaluation metrics.

4.1 Corpora

To evaluate computational models for speech segmentation, it is customary to use transcriptions of real child-directed speech as input data. The first corpus I use in this study is the *BR corpus*, the de-facto standard for evaluating computational models for segmentation. It was originally collected by Bernstein Ratner et al. (1987) by recording the conversations between nine mothers and their children. It makes up part of the English section of CHILDES, a large database that contains orthographic transcriptions of speech between carers and children of a variety of ages in a multitude of languages (MacWhinney and Snow, 1985).

The BR corpus was later hand-processed by Brent and Cartwright (1996) to produce a phoetic transcription, keeping only child-directed utterances and removing onomatopoeia and interjections. They removed all word boundaries, keeping only utterance boundaries, for a total of 95,809 phonemes, 33,387 words and 9,790 utterances. The transcription system used is not standard, often combining diphthongs, r-colored vowels and syllabic consonants into a single symbol. As there are only 50 symbols used, there is an average of 2.9 phonemes per word. Examples of utterances from the corpus can be seen in table 4.1. Segmentation models have the task of correctly placing word boundaries given these input utterances, without any supervision. For example, if the input is 'yuwanttusiD6bUk' then the correct output is 'yu want tu si D6 bUk' (you want to see the book).

In this study, I also evaluate models cross-lingually, using phonetic transcriptions of child-directed speech from twenty-six different languages. The data comes from CHILDES and was processed automatically by Caines et al. (2019) using a speech synthesiser to produce phonetic transcriptions. Each transcription consists of 10,000 utterances of child-directed monolingual speech. These transcriptions use the International Phonetic Alphabet which contains more

Table 4.1: The first five utterances in the BR corpus.

Input utterance	Correct segmentation	Orthographic equivalent
yuwanttusiD6bUk	yu want tu si D6 bUk	you want to see the book
lUkD*z6b7wIThIzh&t	lUk D*z 6 b7 wIT hIz h&t	look there's a boy with his hat
&nd6dOgi	&nd 6 dOgi	and a doggie
yuwanttulUk&tDIs	yu want tu lUk &t DIs	you want to look at this
lUk&tDIs	lUk &t DIs	look at this

symbolic phonemes than the alphabet used for the BR corpus. In these transcriptions, there is an average of 3.7 ± 0.7 phonemes per word due to the more fine-grained phonetic detail. This also varies between languages, for instance the Turkish transcript has an average of 5.4 phonemes per word but the Cantonese transcript only has an average of 2.6 phonemes per word, reminding us that the notion of a 'word' is not equal across languages.

4.2 Evaluation metrics

I report each model's performance standard measures; precision, recall and F_1-score as:

$$P = \frac{TP}{TP + FP}\,, \tag{4.1}$$

$$R = \frac{TP}{TP + FN}\,, \tag{4.2}$$

$$F_1\text{-score} = 2 \times \frac{P \times R}{P + R}\,. \tag{4.3}$$

TP is the number of true positives identified by the model, FP is the number of false positives (items identified by the model that are incorrect with respect to the gold standard) and FN is the number of false negatives (items missed by the model). The F_1-score is calculated as the harmonic mean of precision and recall, providing a single balanced measure. As is conventional, I report F_1-scores as percentages.

Studies of computational segmentation report these measures in three different ways, *boundary*, *token* and *type*:

Boundary scores

TP, FP and FN are calculated according to the boundaries placed. For instance, TP is the number of correctly identified boundaries. This gives BP, BR and BF for the *boundary precision*, *boundary recall* and *boundary F_1-score*. Note that utterance boundaries are not included in these calculations, as these are assumed to be trivial to place.

Token/word scores

A stricter measures that indicate how well word tokens have been identified in the speech stream. As such, true positives are counted only if both boundaries of a word are found without an intervening boundary between them. These scores are necessarily lower than the boundary scores. This gives WP, WR and WF for the *word precision*, *word recall* and *word F_1-score*. Note that these include utterance-initial and utterance-final words.

Type/lexicon scores

These are similar to word scores, but true positives are marked over word types rather than word tokens, so are not skewed by the frequency of each type. This is done by comparing the final lexicon learned by the model to the expected lexicon, the true set of word types in the corpus. If the model is better at segmenting high-frequency words, the lexicon scores will be lower than the word scores. The lexicon scores are LP, LR and LF for the *lexicon precision*, *lexicon recall* and *lexicon F_1-score*.

Finally, I report two error measures to give insight into how the models may fail, related to two of the error types a model may make. First, a model may miss a boundary, causing *under-segmentation*. Second, the model may place a boundary where there should not be one, causing *over-segmentation*. As the simple error counts will change depending on the size of the corpus and as there are many more word-internal positions than boundaries, normalised measures are used for under-segmentation (E_u) and over-segmentation (E_o):

$$E_u = \frac{FN}{FP + TP}\,, \tag{4.4}$$

$$E_o = \frac{FP}{FP + TN}\,, \tag{4.5}$$

where TP, FP and FN are the quantities used for the boundary measures and TN gives the true negatives (the total count of correctly placed word-internal positions). Intuitively, E_u gives the fraction of boundaries marked as word internal and E_o gives the fraction of word internal positions incorrectly marked as boundaries.

4.3 Evaluation procedure

In the machine learning literature, models are typically evaluated by training them on one section of the corpus then testing them on another. Computational models for segmentation, however, are unsupervised. Thus, studies will typically train their models on a single run of the whole corpus and report the average scores across this corpus, without a test set. Although models improve as they learn, they cannot correct past mistakes, usually resulting in average scores lower than if a test-train split were used. This is the evaluation procedure I follow.

As a baseline model, I implemented BASELINE, which assigns boundaries randomly but with the correct probability (the true proportion of word boundaries). BASELINE is therefore more informed than a truly random classifier, as this probability is difficult to estimate. This has been

the customary baseline for evaluating segmentation models since Brent and Cartwright (1996).

Most studies also only report their results on one run of the corpus. As performance may depend on the exact ordering of the utterances in the corpus, I also report results averaged over multiple shuffles of the corpus used, making it clear where I do so. All models reported here are deterministic, so only one run is needed per shuffle.

Chapter 5

Results

In this chapter, I first evaluate my re-implementations of MULTICUE and PHOCUS. I compare older sets of cues to the new set of cues used in MULTICUE-21 and also examine the new variants of the weighted majority-vote algorithm. I then evaluate DYMULTI against PHOCUS and MULTICUE by comparing the average performance, learning rates across the BR corpus and different values of the lexical recognition constant α. I then compare DYMULTI to previous studies to place its performance in context. Finally, I perform cross-lingual evaluation, comparing DYMULTI, PHOCUS and MULTICUE on 26 different languages. This marks the first time that state-of-the-art segmentation models have been compared on so many languages.

In the analysis below, *significance* is calculated using paired sample t-test at significance level $\alpha = 0.01$.

5.1 Segmentation performance of reimplemented PHOCUS and MULTICUE models

5.1.1 Evaluating MULTICUE models

Table 5.1 gives the results for MULTICUE, run with different sets of indicators. My implementation of MULTICUE-14 and MULTICUE-17 achieves similar scores to Çöltekin and Nerbonne (2014); Çöltekin (2017), with slightly higher error rates than the reported results, but far exceeding the baseline. These differing error rates are likely due to fine-grained implementation differences, such as how probability estimates are calculated and how utterances are internally represented.

Table 5.1 also compares running MULTICUE-14 without the stress cue, as Çöltekin and Nerbonne (2014) found the stress cue to decrease performance. The model achieves slightly lower scores than the published results in this case, with a higher under-segmentation error rate. As such, I can confirm the finding of Çöltekin and Nerbonne (2014), that including the stress cue leads to worse overall performance. Also included are the results of MULTICUE-21, whose

Table 5.1: A comparison of the performance of MULTICUE models run on the BR corpus, with the highest scores and lowest error rates (not including referenced results) in **bold**. Italicised lines give the scores reported for each model in their corresponding studies. MULTICUE-14/S is the MULTICUE-14 model without the stress cue. The models are only run once on the corpus to facilitate direct comparison with the corresponding reported scores.

Model	BP	BR	BF	WP	WR	WF	LP	LR	LF	E_u	E_o
MULTICUE-14	**93.4**	78.0	85.0	80.3	70.9	75.3	27.4	60.9	37.8	21.9	**2.1**
Reference	*92.8*	*75.7*	*83.4*	*78.3*	*68.1*	*72.9*	*26.8*	*62.7*	*37.5*	*24.3*	*2.2*
MULTICUE-14/S	84.3	**88.3**	86.2	73.6	76.1	74.8	38.7	64.8	48.5	**11.7**	6.2
Reference	*83.7*	*91.2*	*87.3*	*74.1*	*78.8*	*76.4*	*43.9*	*67.7*	*53.3*	*8.8*	*6.7*
MULTICUE-17	84.0	87.7	85.8	73.1	75.3	74.2	35.6	66.6	46.4	12.3	6.3
Reference	*84.9*	*88.5*	*86.7*	*74.3*	*76.5*	*75.4*	*38.0*	*67.0*	*48.5*	*11.5*	*5.9*
MULTICUE-21	89.7	87.1	**88.4**	**80.2**	**78.5**	**79.3**	**41.2**	**69.6**	**51.7**	12.9	3.8
BASELINE	27.6	29.5	28.5	12.5	13.1	12.8	6.0	43.4	10.6	70.5	29.3

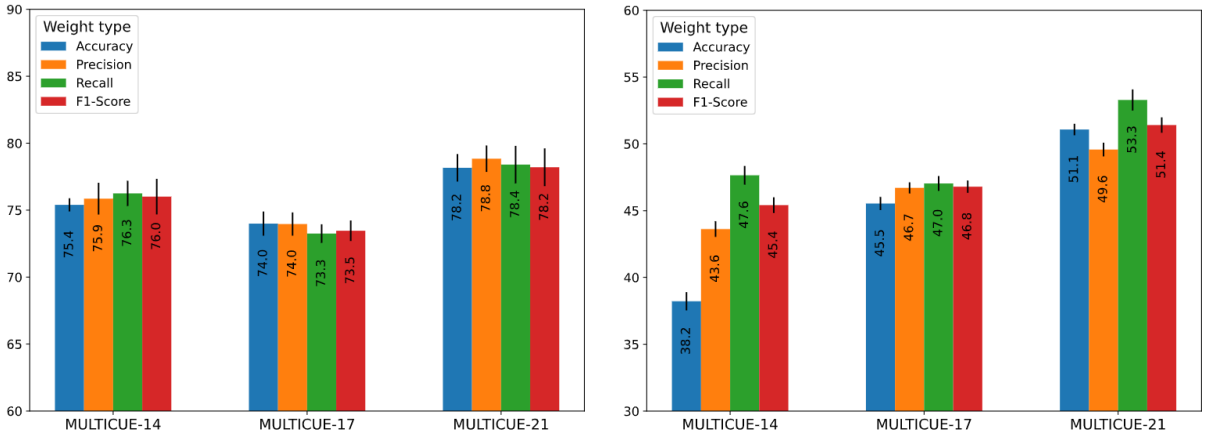
set of indicators combines the strengths of the MULTICUE-14 and MULTICUE-17 models, as described in section 3.1.2. This set of indicators clearly leads to substantial improvements, with MULTICUE-21 achieving better F_1-scores than MULTICUE-14 and MULTICUE-17.

5.1.2 Evaluating new variants of the weighted majority-vote algorithm

In section 3.1.1, I presented the accuracy-based weighted majority-vote algorithm used in MULTICUE and introduced three new variants to that algorithm that instead use weights based on the precision, recall and F_1-score that each indicator achieves for the two labels.

Figure 5.1 presents the WF and LF scores when running the three MULTICUE models using these four weighted majority-vote algorithms. I run these models ten times over different shuffles of the BR corpus to account for ordering effects that may advantage these algorithms. The performance of each weight type varies for each model and each score type, with the recall weights providing the best improvement over the original accuracy weights in most cases. For the LF scores, the performance gain is significant for all three models. The LF scores appear to be much more sensitive to varying the weight type than the WF scores, for which it is just the recall weights for MULTICUE-14 that produces a significant increase. This suggests that the weight type has a larger effect on infrequent words, as WF captures word tokens, so is weighted towards more frequently-occurring words.

The wider distribution of scores across weight types for the MULTICUE-14 and MULTICUE-21 models compared to the MULTICUE-17 model may be related to the fact that MULTICUE-17 only uses predictability cues, whereas the other two models use a range of cues. It may be that these alternative algorithms are more useful when a wide variety of cues are used, as the more nuanced weight schemes may be able to capture the differences between them. Overall, these weight schemes are effective for this task.



(a) Word F_1-scores.

(b) Lexicon F_1-scores.

Figure 5.1: Word and Lexicon F_1-scores (WF, LF) for each model and each weight type, calculated by running each model separately on ten shuffles of the BR corpus and averaging results.

Table 5.2: The performance of the two PHOCUS models on the BR corpus, with the highest scores and lowest error rates in **bold**. Also included are scores for each model averaged over ten shuffles of the BR corpus.

Model	BP	BR	BF	WP	WR	WF	LP	LR	LF	E_u	E_o
PHOCUS-1	82.3	**84.4**	83.3	70.0	71.3	70.7	53.8	55.7	54.7	**15.6**	6.9
PHOCUS-1 (avg)	83.3	79.8	81.5	69.2	67.2	68.2	47.4	54.0	50.5	20.2	6.1
PHOCUS-1S	**91.3**	**84.4**	**87.7**	**81.6**	**77.2**	**79.3**	**57.9**	**67.8**	**62.5**	**15.6**	**3.0**
PHOCUS-1S (avg)	90.9	79.9	85.1	78.8	72.1	75.2	51.7	66.4	58.1	20.1	**3.0**
BASELINE	27.6	29.5	28.5	12.5	13.1	12.8	6.0	43.4	10.6	70.5	29.3

5.1.3 Evaluating PHOCUS models

The performance of the two PHOCUS models is given in table 5.2. Reference rows are not available because Venkataraman (2001) and Blanchard et al. (2010) use different evaluation schema. Venkataraman reports only WP, WR and LP for PHOCUS-1, averaged over 100 shuffles of the corpus, giving 67.7, 70.2 and 52.9 respectively. Averaging over 10 shuffles, I get 69.2 ± 2.9, 67.2 ± 2.5 and 47.4 ± 1.2 respectively. Deriving a WF score from his WP and WR scores gives 68.9, which is close to my WF score of 68.2 ± 2.6.

Blanchard et al. (2010) only give the WF score of PHOCUS-1S, reporting WF = 80, but do not include the first 1000 utterances in this calculation. Replicating this, I achieve a very close WF score of 80.8. Overall, my implementation of these models seems to perform similarly to the original studies.

Comparing the scores achieved by my implementations of the two models, it is clear that the require-syllabic-sound constraint introduced in PHOCUS-1S leads to a significant increase in all F_1-scores, justifying the creation of a boundary-finding model that can use this constraint.

Table 5.3: Comparing the performance of MULTICUE models with DYMULTI models using the same set of indicators, with $\alpha = 0$ for the DYMULTI models. Each model is run on ten shuffles of the BR corpus, averaging scores, with the highest scores and lowest error rates in **bold**.

Model	BP	BR	BF	WP	WR	WF	LP	LR	LF	E_u	E_o
MULTICUE-14	92.8	78.3	84.9	80.0	71.1	75.3	27.6	61.9	38.2	21.7	2.3
DYMULTI-14	**93.9**	80.0	86.4	82.2	73.6	77.6	28.8	62.1	39.4	20.0	**2.0**
MULTICUE-17	83.5	88.1	85.7	72.6	75.4	74.0	34.9	65.6	45.6	11.9	6.6
DYMULTI-17	90.1	**88.4**	89.3	82.0	**80.9**	81.4	38.2	70.2	49.5	**11.6**	3.7
MULTICUE-21	88.4	87.4	87.8	78.5	77.9	78.2	40.3	69.8	51.1	12.6	4.4
DYMULTI-21	92.0	87.0	**89.5**	**83.4**	80.2	**81.8**	**40.8**	**71.2**	**51.9**	13.0	2.8
BASELINE	27.6	29.5	28.5	12.5	13.1	12.8	6.0	43.4	10.6	70.5	29.3

The over-segmentation error rate is halved from 6.1 to 3.0, indicating that this constraint is preventing boundaries being placed at word-internal positions that would otherwise lead to producing words without syllabic sounds.

Table 5.2 also reveals that F_1-scores decrease when the input corpus is shuffled (avg). This indicates that the specific ordering of utterances in the BR corpus is useful for segmentation. As the ordering of utterances comes from real child-directed speech, this suggest that parents may positively bias the ordering of utterances spoken to their children to assist with segmentation, such as pairing new word types with previously-uttered word types.

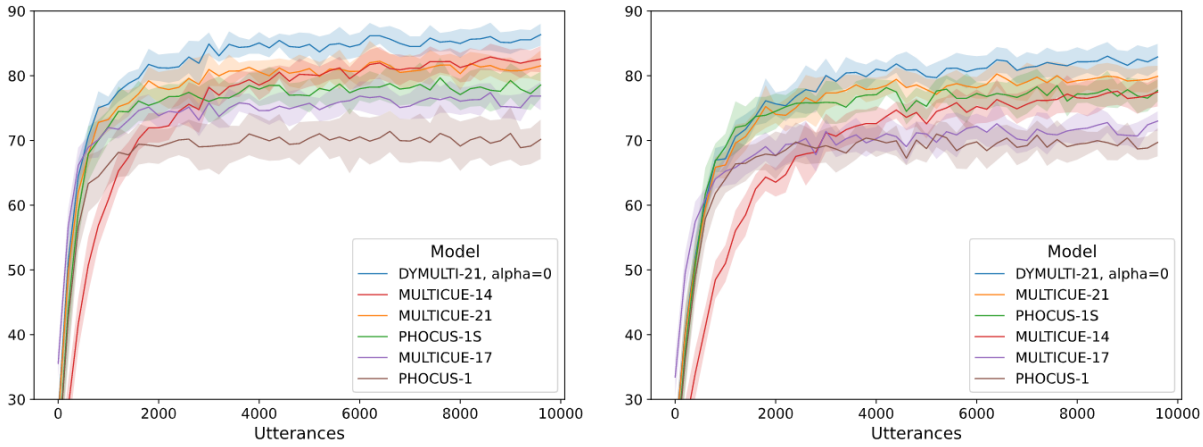
5.2 Segmentation performance of DYMULTI model

5.2.1 Comparing DYMULTI to MULTICUE when using the same set of indicators

Table 5.3 gives the full scores comparing MULTICUE to DYMULTI, considering just the require-syllabic-sound constraint. Every F_1-score is significantly improved using the DYMULTI model, with DYMULTI-21 achieving the best F_1-scores; 89.5, 81.8 and 51.9 for BF, WF and LF respectively. Generally, using the DYMULTI model significantly decreases the over-segmentation error rate of the MULTICUE models. This confirms that the require-syllabic-sound constraint alone is a useful addition, correctly preventing the model from placing erroneous boundaries. The increase in WF and LF scores shows that the Viterbi algorithm, considering this constraint, is finding the correct words to segment, leading to a more accurate lexicon.

5.2.2 Comparing learning rates of segmentation models

The scores in table 5.3 were calculated by taking the average performance of each model over the whole BR corpus, including the many initial mistakes made as the models gather statistical information. While this gives an indication of how well each model learns to segment after



(a) Word F_1-scores.

(b) Lexicon F_1-scores.

Figure 5.2: Word and Lexicon F_1-scores (WF, LF) for a selection of the models implemented in this study, calculated over blocks of 200 utterances. Scores are calculated by running each model separately on ten shuffles of the BR corpus and averaging results.

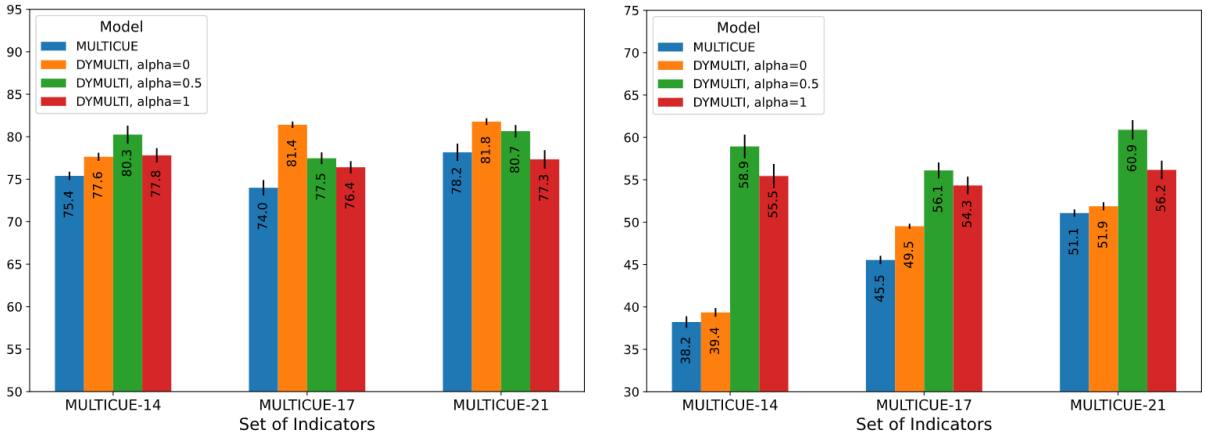
beginning with no knowledge of the target language, it can be more informative to see how the performance of each model progresses across the corpus.

Figure 5.2 gives the WF and LF learning rates for a selection of the models described in this study. These models initially perform very poorly, but quickly improve over the first 1000–2000 utterances, after which scores do not increase or decrease by more than 10 points. This is expected, as the models begin with very poor representations of the target language and so make poor boundary decisions. As such, the average scores over the whole corpus are not representative of the final performance of each model. For example, over the whole corpus, MULTICUE-14 has an average WF score of 75.4 ± 0.5 and MULTICUE-21 has an average WF score of 78.2 ± 1.0, but there is no significant difference between their WF scores on the final block of 200 utterances. It seems that MULTICUE-14 initially learns very slowly, likely due to a large number and variety of indicators that need to be learned.

From these learning rates, we also see the consistent benefit of the require-syllabic-sound constraint. At every stage in the learning process, PHOCUS-1S achieves higher F_1-scores than PHOCUS-1 and DYMULTI-21 achieves higher F_1-scores than MULTICUE-21. Indeed, DYMULTI-21 achieves the highest F_1-scores out of any model at almost every stage of the learning process, achieving WF and LF scores of 86.3 ± 1.7 and 82.9 ± 2.0 respectively on the final 200 utterances of the BR corpus, confirming the validity of this model and the benefit of combining the boundary-finding and language modelling approaches to segmentation.

5.2.3 Evaluating the lexical recognition process

Figure 5.3 compares MULTICUE to DYMULTI, considering three values for the lexical recognition parameter α. In all cases but one, the DYMULTI model performs significantly better than the MULTICUE model when both models use the same set of indicators. Similarly to the



(a) Word F_1-scores.

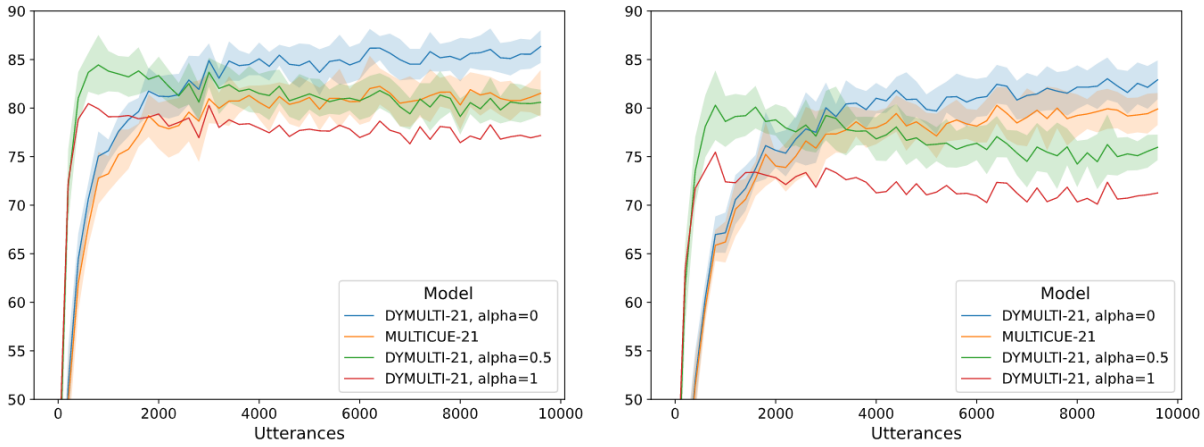
(b) Lexicon F_1-scores.

Figure 5.3: Word and Lexicon F_1-scores (WF, LF) for four models using three sets of indicators. MULTICUE-14, MULTICUE-17 and MULTICUE-21 are compared to three DYMULTI models using the same sets of indicators, setting $\alpha = 0, 0.5, 1$. Scores are calculated by running each model separately on ten shuffles of the BR corpus and averaging results.

majority-vote algorithm variants, it seems the LF scores are more sensitive to this change in model than the WF scores. The LF score for MULTICUE-14, for instance, jumps by more than 20 points when DYMULTI is used with $\alpha = 0.5$. This suggests that the lexical processes of DYMULTI help capture infrequent words.

The learning rates for these models, however, tell a different story. Figure 5.4 compares the learning rates of these models, revealing that the relatively higher *average* WF and LF scores over the corpus for DYMULTI with $\alpha = 0.5$ and $\alpha = 1$ given in figure 5.3 are actually due to a very steep initial learning rate. Indeed, the final WF and LF scores for these two value of α are actually *lower* than DYMULTI-21 with $\alpha = 0$ and in fact lower than MULTICUE-21. It seems that segmenting on the basis of previously-seen words is a useful strategy at the very start of learning to segment, while the boundary cues are still gathering statistical information. As the boundary-based process improves, this lexical recognition procedure actually harms the model, leading to a decrease in WF and LF over time.

I believe that this is because the lexical recognition process described in this study is very rudimentary, simply adding a fixed value to the score when a word is recognised. While the model is gathering statistical information, the boundary-based votes are likely to be inaccurate and extreme, so the lexical recognition process will help prevent incorrect segmentation. As the boundary-based votes become more fine-grained, however, the fixed score of the lexical recognition process will dominate. This will prevent the boundary-based votes from discovering new words, explaining the decrease in LF over time. A more nuanced lexical recognition process should account for this, perhaps by including a decay parameter. Using DYMULTI, it is easy to explore the inclusion of such a lexical process, without needing to define or run a new model.



(a) Word F_1-scores.

(b) Lexicon F_1-scores.

Figure 5.4: Word and Lexicon F_1-scores (WF, LF) for MULTICUE-21 and DYMULTI-21, setting $\alpha = 0, 0.5, 1$. Scores are calculated by running each model separately on ten different shuffles of the BR corpus and averaging results.

5.3 Comparison to previous studies

Table 5.4 compares DYMULTI-21 and MULTICUE-21 to other models in the word segmentation literature. Incremental models are reported, as well as models that train in batch or train on multiple runs of the corpus. DYMULTI-21 with $\alpha = 0$, achieves higher BF and WF scores than all of these, with a comparable LF score. Using $\alpha = 0.5$ results in the highest LF score, but this is potentially misleading, as discussed in the previous section. It is also interesting that DYMULTI-21 outperforms the several-run models of (Fleck, 2008; Ma et al., 2016) and the batch-based models of Goldwater et al. (2009); Elsner and Shain (2017) as these do not suffer from the lower performance in the initial learning phase. It is also worth noting that my implementation of PHOCUS-1S already outperforms most previous studies.

Many previous studies only explore one or two cues for segmentation, stating that they expect a model that considers more cues to perform better (Blanchard et al., 2010; Ma et al., 2016; Çöltekin, 2017). The set of cues chosen here seem particularly effective, with MULTICUE-21 already achieving higher BF and WF scores than previous studies. The LF score for DYMULTI-21 with $\alpha = 0$ is still lower than many of the other models in table 5.4, only outperforming the models of Fleck (2008); Ma et al. (2016); Çöltekin (2017) that do not store a lexicon at all, but setting $\alpha = 0.5$ remedies this, leading to the highest LF score.

This comparison confirms both the strength of the boundary cues included and the strength of the require-syllabic-sound constraint at the lexical-level.

5.4 Cross-lingual evaluation

The majority of studies that present models for child word segmentation only report results on English transcripts, typically only using the BR corpus. Exceptions include Blanchard et al.

Table 5.4: A comparison of MULTICUE-21 and DYMULTI-21 with a variety of top-performing models from the child word segmentation literature. Scores are obtained on the BR corpus, with the highest scores and lowest error rates in **bold**. Also included is PHOCUS-1S, as Blanchard et al. (2010) only report WF. If there were multiple models reported in a study, the model with the highest LF score is given. The scores across models are not always directly comparable, as some are calculated differently to others. The first four models are incremental, so scores are calculated over a single run of the BR corpus (averaging over 100 independent runs over shuffles of the corpus in the case of Venkataraman (2001)). The next two models are incremental, but run over the corpus multiple times and only report the results for the final run. The following two models are batch-based, so scores calculated after many iterations of training over the corpus (ranging from two to several thousand).

Model	BP	BR	BF	WP	WR	WF	LP	LR	LF	E_u	E_o
B (1999)	80.3	84.3	82.3	67.0	69.4	68.2	53.6	51.3	52.4	25.7	-
V (2001)	-	-	-	67.7	70.2	68.9	52.9	-	-	-	-
C & N (2014)	83.7	91.2	87.3	74.1	78.8	76.4	43.9	67.7	53.3	8.8	6.7
C (2017)	84.9	88.5	86.7	74.3	76.5	75.4	38.0	67.0	48.5	11.5	5.9
F (2008)	**94.6**	73.7	82.9	-	-	70.7	-	-	36.6	26.3	-
M, C & H (2016)	-	-	82.9	-	-	68.7	-	-	42.6	17.3	6.4
G, G & J (2009)	90.3	80.8	85.2	75.2	69.6	72.3	63.5	55.2	59.1	19.2	-
E & S (2017)	81	85	83	-	-	72	-	-	-	15	-
PHOCUS-1S	91.3	84.4	87.7	81.6	77.2	79.3	57.9	67.8	62.5	15.6	3.0
MULTICUE-21	89.7	87.1	88.4	80.2	78.5	79.3	41.2	69.6	51.7	12.9	3.8
DYMULTI-21, α=0	92.8	86.4	89.5	**84.4**	80.2	82.2	41.4	**71.4**	52.4	13.6	**2.5**
DYMULTI-21, α=0.5	85.9	**95.7**	**90.6**	79.7	**86.2**	**82.8**	**63.6**	65.3	**64.5**	**4.3**	5.9
BASELINE	27.6	29.5	28.5	12.5	13.1	12.8	6.0	43.4	10.6	70.5	29.3

(2010), who also report results on a Sesotho corpus and Fleck (2008), who also reports results on Spanish and Arabic corpora. As these models are designed to represent the ability of a child to acquire any language, proper evaluation is incomplete if the models are not run on a wide variety of languages. Otherwise, the models could be biased towards learning English, and therefore not represent a truly universal language acquisition procedure.

Figure 5.5 compares the LF scores of PHOCUS-1S, MULTICUE-17 and DYMULTI-21 across 26 languages. These transcripts come from the study of Caines et al. (2019), who compared three different segmentation models cross-lingually. These models either perform significantly worse than those presented here, or process the transcripts several times, so are not comparable to these three single-run incremental state-of-the-art models. To account for the initial learning curve which may differ between models and languages, I only report the scores achieved over the last 5000 utterances of each transcript, after the models have stabilised.

The first observation is that for 15 of the 26 languages, DYMULTI-21 achieves a significantly higher LF score than MULTICUE-17 and PHOCUS-1S. Surprisingly, this does not include the English corpus, where PHOCUS-1S achieves a higher LF score than DYMULTI-21. This is a

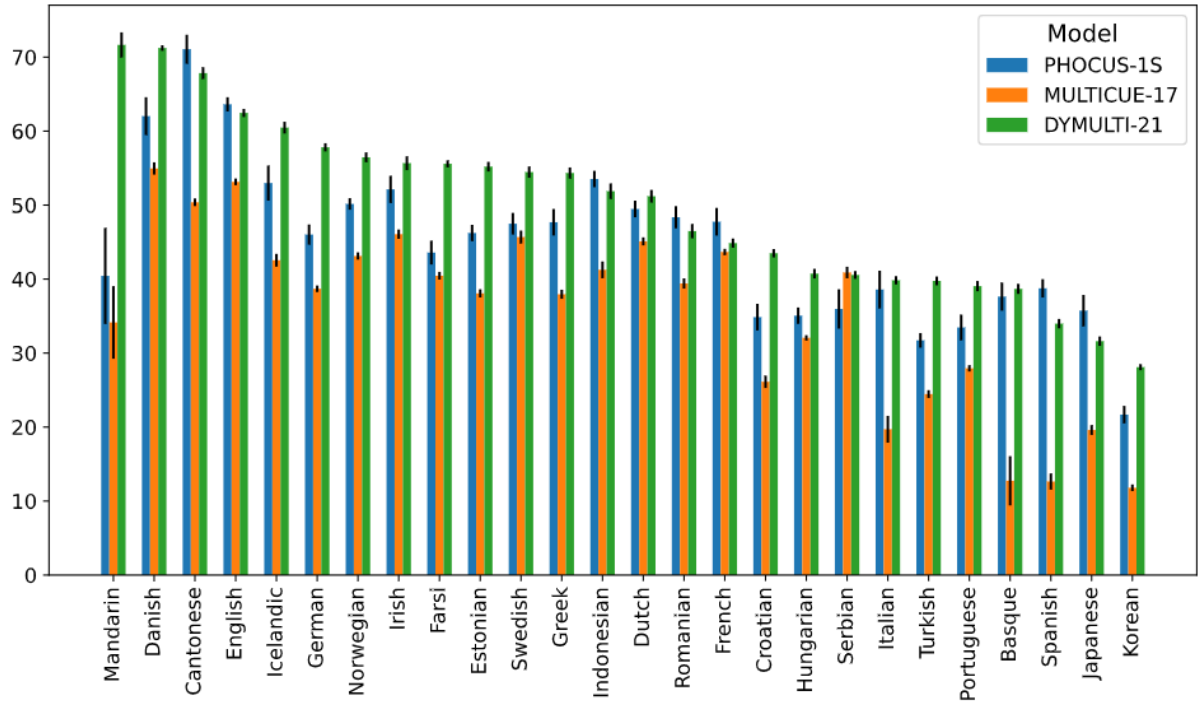


Figure 5.5: Lexicon F_1-scores (LF) for PHOCUS-1S, MULTICUE-17 and DYMULTI-21 with $\alpha = 0$ compared across 26 languages, sorted by LF scores for DYMULTI-21. Scores are calculated by running each model separately on ten shuffles of each transcript and averaging results over the last 5000 utterances of each run, accounting for differing initial learning rates.

different result to the BR corpus, as seen in figure 5.2, where DYMULTI-21 has a consistently higher LF score across all utterances. This indicates the importance of testing on multiple corpora, even for the same language. The 26 transcripts here use the IPA alphabet, which contains many more symbols than the transcription system used for the BR corpus and this clearly has an effect on the performance of the models. Nevertheless, in the majority of cases, DYMULTI-21 outperforms PHOCUS-1S.

Ordering the languages by LF score, English comes second for MULTICUE-17 and PHOCUS-1S and fourth for DYMULTI-21. As DYMULTI builds on MULTICUE and PHOCUS, which themselves build on previous work, this confirms that the research into child segmentation has been biased towards performance on English corpora.

5.5 Summary

In this chapter, I evaluated PHOCUS, MULTICUE and the new DYMULTI framework. My re-implementations of PHOCUS and MULTICUE achieved similar performance to their original studies, successfully replicating their findings. I also found that the cues in MULTICUE-21 and the recall-based weights in my novel weighted majority-vote algorithm both significantly improve the performance of MULTICUE.

Comparing DYMULTI to PHOCUS and MULTICUE, I found that the require-syllabic-constraint significantly improved performance over MULTICUE models using the same set of cues. The lexical recognition process also improved performance, but this was due to a very fast initial learning period; performance actually decreased over time when α was set to 0.5 or 1. DYMULTI also outperformed prior work, including models that used batch training or other weaker constraints.

Finally, I performed cross-lingual evaluation, which has not been done at this scale for state-of-the-art segmentation models. This validated the performance of DYMULTI-21, which outperformed PHOCUS-1S and MULTICUE-17 on 15 of 26 languages.

Chapter 6

Discussion and Summary

In this study, I explored both boundary-finding and language modelling methods for word segmentation, producing a new segmentation framework, DYMULTI, that combines the powerful boundary decisions from the MULTICUE framework of Çöltekin and Nerbonne (2014) with the lexical constraints of the PHOCUS-1S model of Blanchard et al. (2010). In this chapter, I first consider the performance of DYMULTI with respect to the different views of speech processing. I then discuss my re-implementations of PHOCUS and MULTICUE and the novel benchmarking that I have carried out in this study. Finally, I describe the limitations of this study and future directions, concluding with the wider implications of this research.

6.1 Performance of DYMULTI

In section 2.1, I presented two views of speech processing from the cognitive science literature. The interactionist view states that speech segmentation is driven by lexical recognition. The explicit view states that segmentation is purely a result of placing boundaries using sub-lexical information, without making use of any lexical influences.

My goal was to compare the language-modelling and boundary-finding approaches to solving the speech segmentation problem (which relate to these views of speech processing) and to establish if combining these approaches would lead to improvements in performance on transcriptions of child-directed speech. To achieve this, I first reimplemented the PHOCUS models of Venkataraman (2001); Blanchard et al. (2010) and the MULTICUE models of Çöltekin and Nerbonne (2014); Çöltekin (2017). Both models can make use of both sub-lexical and lexical information, but PHOCUS primarily uses lexical cues and MULTICUE primarily uses sub-lexical cues. As such, these models confirm the studies in the cognitive science literature that find that children make use of sub-lexical and lexical cues for solving the word segmentation problem.

Until this study, it was not possible to conclude whether sub-lexical and lexical cues were complementary or alternative explanations for segmentation, as no model had been designed that was able to efficiently combine lexical and sub-lexical cues without constraint. Presenting

the DYMULTI framework, I confirmed an improvement over PHOCUS and MULTICUE models. This implies that sub-lexical and lexical cues are indeed complementary, and that both can be helpful for solving the word segmentation problem.

I also found that DYMULTI-21 outperforms previous state-of-the-art systems for segmentation on the BR corpus, including those that run over the corpus several times or learn in batch. As DYMULTI makes use of cues found to be used by infants for speech segmentation and builds on established, state-of-the-art models, this means that DYMULTI is a good representation of how children may learn to segment speech and begin to build their lexicon.

6.2 Novel benchmarking of state-of-the-art models

Besides presenting a new state-of-the-art model for segmentation, a major contribution of this study was the thorough benchmarking and replication of the PHOCUS and MULTICUE frameworks. Replication is an important scientific discipline and few state-of-the-art models have been re-implemented in prior work. Re-implementing these frameworks, I achieved comparable results to their respective studies. Running MULTICUE-14 without the stress cue, I confirmed the result of Çöltekin and Nerbonne (2014) that it increased performance. I also validated the finding of (Blanchard et al., 2010) that the require-syllabic-sound constraint improves performance, using this as the core motivation for the design of DYMULTI.

Despite most studies in this field using the same corpus for evaluation, they all evaluate their models differently, making cross-comparison difficult. This is also the case for the PHOCUS and MULTICUE frameworks. For example, the PHOCUS studies do not provide boundary scores, and the MULTICUE studies do not provide the learning rates of their models. In this study, I compared these models with a wide range of experiments, including calculating average scores over the whole corpus, plotting learning rates over time and performing novel cross-lingual evaluation. This is the first time these models have been directly compared, producing a useful survey of the field. The cross-lingual evaluation is particularly noteworthy, as few state-of-the-art models have previously been compared on more than two languages. This needs to become a regular practice if the goal of these models is truly to understand how *any* language is acquired, not just English.

6.3 Limitations of child-directed corpora

One of the strengths of the DYMULTI framework is that it is much more flexible than previous models, as it can easily consider multiple boundary-based cues and lexical processes. In this study, this allowed for the combination of phonotactic, utterance-boundary, lexicon and stress cues derived from phonetic transcriptions of child-directed speech.

The BR corpus is the de-facto standard for evaluating such cognitive models, but it has certain limitations. Containing only 9,790 utterances spoken by only nine speakers in 1987, it may not

be fully representative of English child-directed speech. To validate my results, I ran PHOCUS, MULTICUE and DYMULTI on corpora from 26 different languages. These phonemised transcripts also have limitations, having been produced automatically, so may be subject to translation error and bias.

Through this cross-lingual evaluation, I found that the models perform consistently better on English than on most other languages. This is another limitation of using a single corpus for cross-comparison, as it suggests that the entire field may have 'over-fit' to the BR corpus, producing models that perform well on English at the expense of other languages. Using the DYMULTI framework, this could be investigated in future work by experimenting with different cues and implementing different lexical processes to see how these choices alter the performance across languages.

Another limitation of these child-directed speech corpora is that they contain only symbolic phonemes, abstracting away from many other cues available to children for segmentation, such as allophonic variation and realistic stress information. If a corpus containing continuous audio were used, these cues could be extracted. The CAREGIVER corpus (Altosaar et al., 2010) initially seemed like a good candidate for this, containing phonetic transcriptions of child-directed speech aligned with raw audio. Unfortunately the utterances are scripted. As a result, the type-token ratio is only 0.002, much smaller than the 0.039 for the BR corpus. In initial experiments I found that segmentation models that rely on seeing a variety of phoneme combinations at boundaries struggle, resulting in very different results than when run on real child-directed utterances. When a suitable corpus is available, however, DYMULTI should easily be able to incorporate such features either at the boundary-level or at the lexical-level.

6.4 Future work

There are many components of DYMULTI that could be expanded upon in future work. In this study, I proposed a new weighted majority-vote algorithm for cue combination, finding improvements over accuracy-based weights. Bayesian cue combination could be an alternative approach, and indeed it has been used to model other cognitive processes (Colombo and Hartmann, 2017).

The representation of utterances could also be improved. Instead of individual symbols, phonemes could be represented by features, such as the 11 phonetic features used in the model of Christiansen et al. (1998). Language-specific, distributed representations could also replace phonemes, such as the learned acoustic embedding vectors used in the models of Ma et al. (2016) and Kamper et al. (2016).

In this study, I used DYMULTI to explore how infants learn to segment speech. Using her model, Fleck (2008) explored how infant speech segmentation could be upgraded to adult speech segmentation. She did this by introducing a simple syntactic process to prevent affixes from being segmented away from their stems, achieving WF and LF scores of 80.3 and 41.5 respectively

on the Buckeye corpus (Pitt et al., 2005). In initial experiments, DYMULTI performs worse than Fleck's model, with WF and LF scores of 69.9 and 40.6 respectively. Fleck hypothesised that models for infant speech segmentation may be segmenting morphemes, rather than words. Further work into DYMULTI could implement a syntactic process such as Fleck's to investigate this claim.

Finally, as suitable corpora become available, the number and quality of input cues can be improved. Future work could also investigate semantic and multi-modal information that parents may provide their children, such as deictic gesture towards images, joint attention on entities in the environment or iconic gesture to demonstrate object shapes. It is likely that with more cues available, performance would increase, improving our understanding of language acquisition. Nevertheless, the research conducted here has confirmed that a model based only on statistical sequence analysis can successfully bootstrap segmentation across a range of languages.

6.5 Conclusion

In this study, I presented the word segmentation problem and compared two state-of-the-art models for speech segmentation; the PHOCUS models of Venkataraman (2001) and Blanchard et al. (2010), a language modelling approach to speech segmentation, and the MULTICUE models of Çöltekin and Nerbonne (2014), a boundary-finding approach. By re-implementing both models, I found that MULTICUE lacked the ability to consider lexical-level processes and PHOCUS lacked the ability to combine information from several cues. I created a new model, DYMULTI, which overcomes both drawbacks by using the boundary decisions of MULTICUE and converting them to scores that can be passed to the Viterbi algorithm of PHOCUS. In doing so, I achieved state-of-the-art performance on the BR corpus. Evaluating the model cross-lingually, I found that DYMULTI outperformed MULTICUE and PHOCUS on 15 of 26 child-directed speech corpora from different languages, but also that all three models achieved close to their best performance on English, suggesting possible research bias.

I have also demonstrated how unsupervised machine learning techniques can be used to model human behaviour and learning. Through this study into language acquisition, I presented a new algorithm for unsupervised weighted majority voting, by using precision, recall and F_1-score weights for indicators. This algorithm could be incorporated into other machine learning models, outside the field of cognitive modelling.

DYMULTI represents a flexible framework for exploring hypotheses related to the word segmentation problem, efficiently combining lexical and sub-lexical cues. In this study I have explored predictability, utterance and stress as sub-lexical cues, and a require-syllabic-sound constraint and lexical recognition as lexical cues. Such cues could be enhanced with updated knowledge about infant speech cognition to produce a more comprehensive model for speech segmentation. The framework can also be upgraded by considering more nuanced representations of utterances, alternative cue-combination algorithms and other cues for segmentation, once suitable corpora are available. In its current form, it is already the most comprehensive model for infant

speech segmentation to date.

The impacts of DYMULTI and future research into child speech segmentation are plentiful. Using DYMULTI's cue combination system, we can better our understanding of which cues are relevant to segmentation, aiding segmentation in speech recognition models. Adult speech segmentation could also be researched using DYMULTI, by examining which cues are relevant by testing on adult-directed speech corpora and adapting DYMULTI with new grammatical processes accordingly. Finally, this research has contributed to the ever-growing understanding of language acquisition. By designing segmentation models that perform well on child-directed speech, we can learn how children first solve this task, thereby improving how we teach language to children in the first place and how language disorders can be mitigated.

Appendix A

Cues and boundary indicators of MULTICUE

In section 3.1.2, I gave a brief overview of the range of indicators implemented in the MULTICUE models of Çöltekin and Nerbonne (2014); Çöltekin (2017). In this chapter, I describe these cues in detail.

A.1 Predictability statistics

The indicators based on predictability statistics follow the transitional probability studies of Saffran et al.. In Çöltekin and Nerbonne (2014), two different measures of predictability are used, *pointwise mutual information* (MI) and *boundary entropy* (H). Pointwise mutual information is a measure of association defined as

$$MI(l,r) = \log_2 \frac{P(l,r)}{P(l)P(r)}, \tag{A.1}$$

where l and r are the left and right contexts around the inter-phoneme position (initially one phoneme) and $P(l,r)$ is the joint probability of observing lr together in sequence. Boundary entropy is defined as

$$H(l) = -\sum_{r \in A} P(r|l) \log_2(P(r|l)), \tag{A.2}$$

using the same notation, with $P(r|l)$ as the conditional probability of the context r occurring directly after the context l and A is the alphabet of phonemes. Çöltekin and Nerbonne add a pair of indicators (for the partial peak strategy) for each of these measures, motivated by a previous study that found that combining multiple measures of predictability resulted in better segmentation (Coltekin, 2011). They also include indicators using the reverse entropy measure,

$H(r)$, defined as

$$H(r) = -\sum_{l \in A} P(l|r) \log_2(P(l|r)), \tag{A.3}$$

based on the finding that infants are also sensitive to reverse transitional probabilities (Pelucchi et al., 2009). Mutual information does not have a reverse measure, as it is symmetrical. Finally, they also include another set of indicators based on these measures where the contexts l or r are set to three phonemes. This is to capture regularities above the phoneme-level, such as between syllables or words and results in a total of 14 indicators[1]. The probabilities $P(l)$, $P(r)$, $P(r|l)$, $P(l|r)$ and $P(l,r)$ are all estimated from previously-segmented utterances by keeping track of n-gram phoneme counts as the corpus is processed. As these counts are all initially 0, this represents the 'starting with no knowledge' constraint for segmentation models.

A later study by Çöltekin (2017) builds a MULTICUE model using just indicators based on predictability cues. He compares and contrasts boundary entropy, mutual information, transitional probability and successor variety as measures of predictability. Transitional probability is simply the conditional probability of the two contexts, given by

$$TP(l,r) = \frac{P(l,r)}{P(l)_.}, \tag{A.4}$$

Successor variety is the predictability measure originally described by Harris (1955) as being a good measure for predicting morpheme boundaries:

$$SV(l,r) = \sum_{r \in A} c(l,r), \tag{A.5}$$

where

$$c(l,r) = \begin{cases} 1 & \text{if substring } lr \text{ occurs in the corpus,} \\ 0 & \text{otherwise.} \end{cases} \tag{A.6}$$

See Çöltekin (2017) for a full comparison of these four measures. He builds separate MULTICUE models to explore each of these measures in turn, including both the forward and reverse measure and including cues for each length of l or r from one to four phonemes, for a total of 16 cues[2], finding that this strategy can produce results comparable to the state-of-the-art language-modelling algorithms of Brent (1999); Venkataraman (2001); Blanchard et al. (2010). He chooses four as the maximum context as smaller or lower values lead to lower performance.

[1] There is pair of indicators (for the partial-peak strategy) for each of $MI(l,r)$, $H(l)$ and $H(r)$ and with $|l| = 1$ and $|r| = 1$, totalling 6. There is also a pair of indicators for each of $MI(l,r)$ and $H(l)$ with $|l| = 3$ and $|r| = 1$ for another 4. Finally, there is a pair of indicators for each of $MI(l,r)$ and $H(r)$ with $|l| = 1$ and $|r| = 3$ for a final 4. Both contexts are never more than length 1 at once.

[2] There are 4 pairs of cues for the forward measure and 4 pairs of cues for the reverse measure.

A.2 Utterance boundaries

Just as infants can make use of predictability measures at boundaries for segmentation, they can also make use of learned phonotactic constraints. Utterance boundaries provide a means to learn these phonotactic constraints, by considering them as noisy word boundaries. They are noisy as the distribution of phonemes around utterance boundaries will differ from the distribution of phonemes around word boundaries due to language's grammar defining word ordering. Nevertheless, they can still provide useful estimates for word-initial and word-final phoneme clusters. Most computational models use utterance boundaries implicitly by assuming that words cannot pass over these boundaries. Other models use utterance boundaries explicitly, such as in Christiansen et al. (1998) and Fleck (2008).

In Çöltekin and Nerbonne (2014), the probabilities of utterance beginnings and utterance endings, $P(ub|r)$ and $P(ub|l)$ respectively, are used as indicators for word boundaries. These probabilities are estimated from the previously-seen utterances and the 'partial peaks' strategy is used to suggest boundaries. Just as with the predictability cues, l and r are also varied to consider contexts of one or three phonemes, for a total of eight indicators.

A.3 Lexicon

Çöltekin and Nerbonne (2014) also derive two sets of indicators from the lexicon. The lexicon is populated with words from previously-segmented utterances, also storing frequency counts. The inclusion of these cues means that the full model no longer follows the explicit view of speech processing, as lexical information is used, but the model is still boundary-finding as the goal is segmentation, not lexical recognition.

The first set of indicators comes from matching word-like strings on either side of a boundary. Given a boundary location, the frequencies of already known word beginnings or endings at that position are counted by matching the substring starting or ending at that boundary to all the words in the lexicon. One pair of indicators does this for word endings (looking at the left context) and another pair does the same for word beginnings (looking at the right context), using the usual 'partial peak' strategy to suggest boundaries for a total of four indicators.

The second set of indicators is identical to the utterance boundary indicators, but the phonotactic constraints are learnt from the word types in the lexicon, rather than from utterance boundaries. Given that utterance boundaries are noisy, this should mean that this cue is more accurate, in that it learns from word boundaries, although these are not guaranteed to be accurate, especially at the start of the process as the model is still very inaccurate. This provides another eight indicators to the full model.

A.4 Lexical stress

Finally, Çöltekin and Nerbonne (2014) define a set of indicators based on lexical stress. Lexical stress is a relatively understudied cue in the computational studies, mostly due to a lack of corpora with realistic stress alignment. For their study, they mark a corpus with lexical stress semi-automatically following the procedure of Christiansen et al. (1998).

To use lexical stress as a cue, they estimate $P(ub|l)$ and $P(ub|r)$, similarly to the lexical utterance boundary cue, but using stress patterns for l and r instead of phonemes. Once again, indicators are produced for contexts of length one and three and a pair is created for each for the 'partial peak' strategy, for a total of eight indicators.

Appendix B

Algorithms

B.1 Full algorithm for PHOCUS

The full algorithm for PHOCUS-1S is given in listings 1 and 2. The `segment_utterance` function processes each utterance using the Viterbi algorithm. A grid of scores is produced by iterating over every bisection of every prefix of the utterance to find the best positions to place word boundaries. Since the score for each segmentation is given by the negative log probability, the function finds the lowest score. Not included in listing 1 is the update procedure. After the utterance is segmented, each segmented word is added to the lexicon, increasing frequency counts. The frequency of each phoneme present in the utterance is also increased.

The lexicon and phoneme frequencies are used in the `word_score` function, which estimates $-\log P(w)$ for a given word w. This function follows directly from the language model given by equations (3.15) and (3.16). The require-syllabic-sound constraint is implemented by returning a probability of 0 if the word does not contain a syllabic sound. In practice, as $-\log(0)$ is not defined, I return 10000 if this or any other probability gives 0. To remove the constraint, `blanchard_model` just needs to be set to `False`, giving PHOCUS-1.

B.2 Viterbi algorithm for DYMULTI

The full algorithm for DYMULTI is given in listing 3. Not included is the update procedure. After each utterance is segmented, each word in the segmentation is added to the lexicon, increasing frequency counts. The frequency of each phoneme present in the utterance is also increased.

The full `word_score()` function for DYMULTI is given in listing 4. This function combines the boundary score given by the weighted majority-vote, the syllabic constraint and lexical recognition process. Removing lines 3-8 results in a model that performs identically to MULTICUE.

```
function segment_utterance(utterance):
    n = len(utterance)
    best_scores = []
    best_segpoints = []

    # Get the best segmentations for each sub-utterance
    for i from 0 to n+1:
        best_score := word_score(utterance[0..i])
        best_segpoint := i

        for j from 1 to i-1:
            word := utterance[j..i]
            score := best_scores[j] + word_score(word)
            if score < best_score:
                best_score := score
                best_segpoint := j
        best_scores.append(best_score)
        best_segpoints.append(best_segpoint)

    # Reconstruct best segmentation
    segpoint = n
    while segpoint != 0:
        new_segpoint := best_segpoints[segpoint]
        if new_segpoint = segpoint then break
        utterance.place_boundary(new_segpoint)
        segpoint := new_segpoint

    return utterance
```

Listing 1: The iterative Viterbi algorithm used to find the best segmentation an utterance. The algorithm builds a grid of scores corresponding to possible segmentations then builds the best segmentation by following the pointers stored in `best_segpoints`. Note that the best score chosen by the algorithm is actually the smallest, as scores are negative log probabilities which grow as the probability shrinks.

```
function word_score(word):
    if not word in LL
        if blanchard_model and not has_syllabic_sound(word):
            return -log(0)
        unseen_prob := L.type_count / (L.type_count + L.token_count)
        boundary_prob := P.relative_freq('#')
        score := -log(unseen_prob) -log(boundary_prob/(1-boundary_prob))
        for phoneme in word:
            score := score - log(P.relative_freq(phoneme))
    else:
        word_prob := L.freq(word) / (L.type_count + L.token_count)
        score := -log(word_prob)
    return score
```

Listing 2: The function used to calculate $-\log P(w)$ for a word w, using the 1-gram language model. For previously seen words, it is given by the word's relative frequency, equation (3.15). For unseen words, the probability is estimated by the frequencies of the phonemes in the word, equation (3.16). L is the lexicon and P is an object that stores phoneme probabilities. Both are updated after each segmentation. For the Blanchard extension, a probability of 0 is returned for words that do not have a syllabic sound.

```
function segment_utterance(utterance):
    n = len(utterance)
    best_scores = []
    best_segpoints = []

    # Get the best segmentations for each sub-utterance
    for i from 0 to n+1:
        best_score := word_score(utterance[0..i])
        best_segpoint := i

        for j from 1 to i-1:
            word := utterance[j..i]
            score := best_scores[j] + word_score(word , score(j) )
            if score > best_score:
                best_score := score
                best_segpoint := j
        best_scores.append(best_score)
        best_segpoints.append(best_segpoint)

    # Reconstruct best segmentation
    segpoint = n
    while segpoint != 0:
        new_segpoint := best_segpoints[segpoint]
        if new_segpoint = segpoint then break
        utterance.place_boundary(new_segpoint)
        segpoint := new_segpoint

    return utterance
```

Listing 3: The adjusted iterative Viterbi algorithm used in DYMULTI, with differences to the PHOCUS model highlighted. The algorithm finds the best segmentation by maximising the sum of the scores for each word in the segmentation, where scores are given by the updated `word_score()` seen in listing 4.

```
function word_score(word, boundary_score):
    score := boundary_score
    # Syllabic constraint
    if has_syllabic_sound(word):
        return -100
    # Lexical recognition process
    if word in L:
        score := score + ALPHA
    return score
```

Listing 4: The function used to calculate the score of `word` for the DYMULTI-L model, where `boundary_score` is the score assigned to the boundary left of the word. The syllabic constraint rejects words without syllabic sounds, as in the `word_score()` function for PHOCUS given in listing 2. The lexical recognition process increases the score of previously-seen words by the model's parameter, `ALPHA`, where L is the proto-lexicon.

Bibliography

Altosaar, T., ten Bosch, L., Aimetti, G., Koniaris, C., Demuynck, K., and Heuvel, H. (2010). A speech corpus for modeling language acquisition: Caregiver.

Aslin, R. N., Saffran, J. R., and Newport, E. L. (1998). Computation of conditional probability statistics by 8-month-old infants. *Psychological Science*, 9(4):321–324.

Bernard, M. (2018). bootphon/wordseg: wordseg-0.7.1.

Bernstein Ratner, N., Nelson, K., and van Kleeck, A. (1987). Children's language.

Blanchard, D., Heinz, J., and Golinkoff, R. (2010). Modeling the contribution of phonotactic cues to the problem of word segmentation. *Journal of Child Language*, 37(3):487–511.

Bortfeld, H., Morgan, J. L., Golinkoff, R. M., and Rathbun, K. (2005). Mommy and me: Familiar names help launch babies into speech-stream segmentation. *Psychological Science*, 16(4):298–304.

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

Brent, M. R. and Cartwright, T. A. (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61(1-2):93–125.

Brent, M. R. and Siskind, J. M. (2001). The role of exposure to isolated words in early vocabulary development. *Cognition*, 81(2):B33–B44.

Caines, A., Altmann-Richer, E., and Buttery, P. (2019). The cross-linguistic performance of word segmentation models over time. *Journal of Child Language*, 46(6):1169–1201.

Cairns, P., Shillcock, R., Chater, N., and Levy, J. (1994). Lexical segmentation: The role of sequential statistics in supervised and un-supervised models. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society: Atlanta, Georgia, 1994*, pages 136–141. Routledge.

Cairns, P., Shillcock, R., Chater, N., and Levy, J. (1997). Bootstrapping word boundaries: A bottom-up corpus-based approach to speech segmentation. *Cognitive Psychology*, 33(2):111–153.

Christiansen, M. H., Allen, J., and Seidenberg, M. S. (1998). Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*, 13(2-3):221–268.

Cole, R. and Jakimik, J. (1978). How words are heard. *Strategies of information processing*, pages 67–117.

Cole, R. A. and Jakimik, J. (1980). A model of speech perception. *Perception and production of fluent speech*, 133(64):133–42.

Colombo, M. and Hartmann, S. (2017). Bayesian cognitive science, unification, and explanation. *The British Journal for the Philosophy of Science*, 68(2):451–484.

Coltekin, C. (2011). *Catching words in a stream of speach: computational simulations of segmenting transcribed child-directed speech*. PhD thesis. Relation: http://www.rug.nl/ Rights: University of Groningen 2011/c.coltekin/pub001 ISBN: 978-90-367-5259-6 (electronic), 978-90-367-5232-9 (print) Research Institute: CLCG, University of Groningen.

Çöltekin, Ç. (2017). Using Predictability for Lexical Segmentation. *Cognitive Science*, 41(7):1988–2021.

Çöltekin, Ç. and Nerbonne, J. (2014). An explicit statistical model of learning lexical segmentation using multiple cues. In *Proceedings of the 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)*, pages 19–28.

Cooper, W. E. and Paccia-Cooper, J. (1980). *Syntax and Speech*. Number 3. Harvard University Press.

Cutler, A. (1996). Prosody and the word boundary problem.

Cutler, A. and Carter, D. M. (1987). The predominance of strong initial syllables in the English vocabulary. *Computer Speech and Language*, 2(3-4):133–142.

Cutler, A. and Mehler, J. (1993). The periodicity bias. *Journal of Phonetics*, 21(1-2):103–108.

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

Elsner, M. and Shain, C. (2017). Speech segmentation with a neural encoder model of working memory. In *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 1070–1080. Association for Computational Linguistics (ACL).

Feng, H., Chen, K., Deng, X., and Zheng, W. (2004). Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Fleck, M. M. (2008). Lexicalized phonotactic word segmentation. In *Proceedings of ACL-08: HLT*, pages 130–138.

Gambell, T. and Yang, C. (2006). Word segmentation: Quick but not dirty. *Unpublished manuscript*.

Gleitman, L. R., Gleitman, H., Landau, B., and Wanner, E. (1988). Where learning begins: Initial representations for language learning. *Linguistics: The Cambridge Survey: Volume 3, Language: Psychological and Biological Aspects*, pages 150–193.

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

Greenberg, J. H. and Jenkins, J. J. (1966). Studies in the psychological correlates of the sound system of american english. *Word*, 22(1-3):207–242.

Harris, Z. S. (1955). From phoneme to morpheme. *Language*, 31(2):190–222.

Johnson, E. K. and Jusczyk, P. W. (2001). Word segmentation by 8-month-olds: When speech cues count more than statistics. *Journal of Memory and Language*, 44(4):548–567.

Jusczyk, P. W. (1999). How infants begin to extract words from speech.

Jusczyk, P. W., Cutler, A., and Redanz, N. J. (1993). Infants' Preference for the Predominant Stress Patterns of English Words. *Child Development*, 64(3):675–687.

Jusczyk, P. W. and Hohne, E. A. (1997). Infants' memory for spoken words. *Science*, 277(5334):1984–1986.

Jusczyk, P. W., Hohne, E. A., and Bauman, A. (1999a). Infants' sensitivity to allophonic cues for word segmentation. *Perception and Psychophysics*, 61(8):1465–1476.

Jusczyk, P. W., Houston, D. M., and Newsome, M. (1999b). The Beginnings of Word Segmentation in English-Learning Infants. *Cognitive Psychology*, 39(3-4):159–207.

Kamper, H., Jansen, A., and Goldwater, S. (2016). Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(4):669–679.

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, 108(2):212–261.

Ma, J., Çöltekin, Ç., and Hinrichs, E. (2016). Learning phone embeddings for word segmentation of child-directed speech. In *Proceedings of the 7th Workshop on Cognitive Aspects of Computational Language Learning*, pages 53–63.

MacWhinney, B. and Snow, C. (1985). The child language data exchange system. *Journal of Child Language*.

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information.

Marslen-Wilson, W. D. and Welsh, A. (1978). Processing interactions and lexical access during word recognition in continuous speech. *Cognitive Psychology*, 10(1):29–63.

Mattys, S. L. and Jusczyk, P. W. (2001). Phonotactic cues for segmentation of fluent speech by infants. *Cognition*, 78(2):91–121.

Mattys, S. L., Jusczyk, P. W., Luce, P. A., and Morgan, J. L. (1999). Phonotactic and Prosodic Effects on Word Segmentation in Infants. *Cognitive Psychology*, 38(4):465–494.

McClelland, J. L. and Elman, J. L. (1986). The trace model of speech perception. *Cognitive Psychology*, 18(1):1–86.

Monaghan, P. and Christiansen, M. H. (2010). Words in puddles of sound: Modelling psycholinguistic effects in speech segmentation. *Journal of Child Language*, 37(3):545.

Narasimhamurthy, A. (2005). Theoretical bounds of majority voting performance for a binary classification problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1988–1995.

Ngon, C., Martin, A., Dupoux, E., Cabrol, D., Dutat, M., and Peperkamp, S. (2013). (non) words,(non) words,(non) words: evidence for a protolexicon during the first year of life. *Developmental Science*, 16(1):24–34.

Pelucchi, B., Hay, J. F., and Saffran, J. R. (2009). Learning in reverse: Eight-month-old infants track backward transitional probabilities. *Cognition*, 113(2):244–247.

Pitt, M. A., Johnson, K., Hume, E., Kiesling, S., and Raymond, W. (2005). The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.

Saffran, J. R., Aslin, R. N., and Newport, E. L. (1996a). Statistical cues in language acquisition: Word segmentation by infants. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, pages 376–380. Lawrence Erlbaum Associates Mawah, NJ.

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

Saffran, J. R., Newport, E. L., and Aslin, R. N. (1996c). Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35(4):606–621.

Saksida, A., Langus, A., and Nespor, M. (2017). Co-occurrence statistics as a language-dependent cue for speech segmentation. *Developmental Science*, 20(3):e12390.

Shi, R. and Lepage, M. (2008). The effect of functional morphemes on word segmentation in preverbal infants. *Developmental Science*, 11(3):407–413.

Sproat, R. and Shih, C. (1990). A statistical method for finding word boundaries in chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351.

Suomi, K. (1993). An outline of a developmental model of adult phonological organization and behaviour. *Journal of Phonetics*, 21(1-2):29–60.

Suomi, K., McQueen, J. M., and Cutler, A. (1997). Vowel harmony and speech segmentation in Finnish. *Journal of Memory and Language*, 36(3):422–444.

Thiessen, E. D. and Saffran, J. R. (2003). When cues collide: use of stress and statistical cues to word boundaries by 7-to 9-month-old infants. *Developmental Psychology*, 39(4):706.

Thiessen, E. D. and Saffran, J. R. (2007). Learning to Learn: Infants' Acquisition of Stress-Based Strategies for Word Segmentation. *Language Learning and Development*, 3(1):73–100.

Venkataraman, A. (2001). A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):350–372.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Witten, I. H. and Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094.