Assignment: A Non-parametric Bayesian Model for Word Segmentation

Sander Nugteren
6042023

Ties van Rozendaal
10077391

Joost Baptist
10760105

April 17, 2016

1 Introduction

The problem of how infants learn to identify individual words in spoken language has been the topic of many research projects for some time now. Since words in a sentence are rarely spoken in isolation (i.e. with pauses in between), there is no trivial way for infants to learn where word boundaries are. This has caused researchers to suspect that infants actually employ statistical strategies as a first step in learning word boundaries, in which statistical regularities play an important role [7].

Early work on statistical word segmentation relies on the observation that transitions between syllables or phonemes are generally less predictable at word boundaries than within words ([4], [6]), which can give the learner a cue as to whether or not there should be word boundary. Behavioral research has shown that infants are sensitive to this effect [6] [1]. This gives rise to the assumption that words are units that, to some degree, help predict other units in a sentence.

[2] proposed a Bayesian framework for the statistical word segmentation problem with the goal to identify the assumptions the learner must make in order to correctly segment (real) natural language. They investigate what kind of words learners with different assumptions are able to identify, using a corpus of phonetically transcribed child-directed speech. Specifically, they test the hypothesis that words are statistically independent units, to which end they developed two different models.

The first model is a unigram model: it treats each word independently (i.e. words do not predict later words). They find that this model has a tendency to under-segment the corpus, by identifying frequently co-occuring sequences as a single word. For example, because sequences like *would you* and *that's a* are relatively common, the learner may be tempted to classify these sequences as single words (i.e. *wouldyou* and *thatsa*).

The second model is a bigram model that assumes that words can predict later words. That is, this model assumes that the choice of a word is conditioned on the previous word. Because of this assumption this model is able to greatly reduce the problem of under-segmentation.

Both models in [2] use a (hierarchical) Dirichlet Process (DP) to define probabilities over clustering of word tokens. The Dirichlet process integrates over the number of clusters K, and therefore is useful when K is unknown. The probability of assigning a new data point z_i to an existing cluster k, is proportional to the size of that cluster. The exact probabilities are given by:

$$P(z_i = k|\mathbf{z}_{-i}) = \begin{cases} \frac{n_k^{(\mathbf{z}_{-i})}}{i-1-\alpha}, & 1 \leq k \leq K(\mathbf{z}_{-i}) \\ \frac{\alpha}{i-1-\alpha} & k = K(\mathbf{z}_{-i}) \end{cases} \tag{1}$$

where $\mathbf{z}_{-i}$ are the clusters without data point i, and $n_K^{(\mathbf{z}_{-i})}$ is the number datapoints assigned to the cluster k.

Even though the DP has been found very useful in clustering, the distributions that it prefers do not match the power law distributions that are usually observed over types and tokens. In order to make the DP more suitable for these word distributions, the Pitmann-Yor process (PYP) [5] introduces a discount factor β. The probabilities for clustering become:

$$P(z_i = k|\mathbf{z}_{-i}) = \begin{cases} \frac{n_k^{(\mathbf{z}_{-i})} - \beta}{i-1-\alpha}, & 1 \leq k \leq K(\mathbf{z}_{-i}) \\ \frac{\alpha + \beta K}{i-1-\alpha} & k = K(\mathbf{z}_{-i}) \end{cases} \tag{2}$$

Indeed, the distributions generated by the PYP model have been found to be very close to the empirical distributions over types and tokens [3].

The goal of this project is to reproduce the unigram model by [2] and to replicate their results. We experiment with different initialization strategies and parameter values and show their effect on precision, recall and the F_0-measure.

In a qualitative analysis we show that the unigram model does indeed have a tendency to under-segment the corpus. Finally, we discuss the merits and shortcomings of this model.

We also implement a unigram model based on the PYP process. We compare this model to the DP unigram model, and investigate the effect of different parameters using a grid search. The DP model structurally outperforms the PYP model. We discuss why this happens, and suggest improvements for this model.

2 Model

2.1 PYP Model

2.1.1 Model definitions

The probabilities of the seating according to the PYP are given by (2).

Now we include the label and the base distribution P_0 in the model:

$$\begin{aligned}
p(w_i = \ell | \mathbf{z}_{-i}, \boldsymbol{\ell}(\mathbf{z}_{-i}), \alpha) &= \sum_{k=1}^{K(\mathbf{z}_{-i})} \mathcal{I}(\ell_k = \ell) \frac{n_k^{(\mathbf{z}_{-i})} - \beta}{i-1+\alpha} + P_0(\ell) \frac{\alpha + \beta K}{i - 1 - \alpha} && (3) \\
&= \frac{n_\ell^{(\mathbf{w}_{-i})} - n_\ell^{\boldsymbol{\ell}(\mathbf{z}_{-i})}\beta}{i-1+\alpha} + P_0(\ell)\frac{\alpha+\beta K}{i-1-\alpha} && (4) \\
&= \frac{P_0(\ell)\alpha + n_\ell^{(\mathbf{w}_{-i})} + \beta(K - n_\ell^{\boldsymbol{\ell}(\mathbf{z}_{-i})})}{i-1+\alpha} && (5)
\end{aligned}$$

Where $n_\ell^{(\mathbf{w}_{-i})}$ is the number of times the label ℓ occurs in the segmented corpus and $n_\ell^{\boldsymbol{\ell}(\mathbf{z}_{-i})}$ is the number of tables where label ℓ occurs.

2.1.2 Inference

Because the joint distribution is defined computationally complex, we use Gibbs sampling to find the maximum a posteriori segmentation.

For Gibbs sampling, we pick a possible boundary location, and consider two hypothesis:
h_1: There is no boundary at this location
h_2: There is a boundary at this location

h^- denotes the set of words and seating arrangement shared by the both hypothesis.

The corresponding probabilities are:

$$\begin{aligned}
P(h_1|h^-) &= P(w_1|h^-)P(u_{w_1}|h^-) && (6) \\
&= \frac{P_0(w_1)\alpha + n_{w_1}^{(\mathbf{w}_{-i})} + \beta(K - n_{w_1}^{\boldsymbol{\ell}(\mathbf{z}_{-i})})}{n^- + \alpha} \frac{n_u^{(h^-)} + \frac{\rho}{2}}{n^- + \rho} && (7)
\end{aligned}$$

$$\begin{aligned}
P(h_2|h^-) &= P(w_2, w_3|h^-) && (8) \\
&= P(w_2|h^-)P(u_{w_2}|h^1)P(w_3|w_2, h^-)P(u_{w_3}|u_{w_2}|h^-) && (9) \\
&= \frac{P_0(w_2)\alpha + n_{w_2}^{(\mathbf{w}_{-i})} + \beta(K - n_{w_2}^{\boldsymbol{\ell}(\mathbf{z}_{-i})})}{n^- + \alpha} \frac{n^- - n_{\$}^{(h^-)} + \frac{\rho}{2}}{n^- + \rho} && (10) \\
&\quad \cdot \frac{P_0(w_3)\alpha + n_{w_3}^{(\mathbf{w}_{-i})} + \mathcal{I}(w_2 = w_3)\beta(K - n_{w_3}^{\boldsymbol{\ell}(\mathbf{z}_{-i})})}{n^- + \alpha + 1} && (11) \\
&\quad \cdot \frac{n_u^{(h^-)} + \mathcal{I}(w_2 = w_3) + \frac{\rho}{2}}{n^- + 1 + \rho} && (12)
\end{aligned}$$

For the PYP model, we have to model h^- explicitly, by removing words from the seating arrangement. The algorithm used is algorithm 1 where addCustomer adds customers to a table proportional to (2) and removeCustomer selects one of the tables of the word, proportional to it's count. Both functions will also update K, and remove the tables if they have become empty.

Algorithm 1: Pseudo algorithm

```
Initialization
segmentation ← initialize randomly
seating ← $\emptyset$
for all words $w_i$ in segmentation do
    addCustomer($w_i$)
end
Gibbs Sampling
for all possible boundaries $b_i$ do
    $w_1$ ← word if boundary is not placed (h1)
    $w_2, w_3$ ← word if boundary is placed (h2)
    if boundary $b_i$ currently exists then removeCustomer($w_2$)
    removeCustomer($w_3$) ;
    else removeCustomer($w_1$) ;
    calculate $p(h1)$
    calculate $p(h2)$
    insert_boundary$_i$ ← sample proportionally
    if insert_boundary$_i$ then addCustomer($w_2$) addCustomer($w_3$) ;
    else addCustomer($w_1$) ;
end
```

2.2 DP model

The PYP is an abstraction of the DP, and the corresponding expressions are therefore the same. Cancelling all terms of β (or setting β to 0) results in the expressions we used for the DP model.

The implementation of the DP model however, is significantly different from the PYP model. When integrating over the seating arrangements we can just sum the clusters with the same word. Therefore, there is no need to model h^- exactly, we can instead just substract 1 from the corresponding counts.

As a result, the DP model is very efficient compared to the PYP model

3 Experiments

3.1 Evaluation metrics

Following [2], we define two types of evaluation metrics: the joint probability of the corpus and retrieval measures.

3.1.1 Corpus probability

For the DP model, the joint probability over all the words in the corpus is defined for as follows:

$$p(\mathbf{w}|\alpha, P_0) = \prod_{w_i \in \mathbf{V}} \left(\frac{n_{w_i} - 1 + \alpha P_0(w_i)}{N - 1 + \alpha} \right)^{n_{w_i}} \tag{13}$$

where $\mathbf{V}$ is the lexicon or vocabulary, N is the total number of words in the corpus, and n_{w_i} is the number of occurrences of word w_i in the corpus.

3.1.2 Retrieval measures

We assess the quality of the retrieved segmentation using precision, recall and the F_0 measure. Like [2], we evaluate these measures on words (per utterance), on boundaries (per utterance, excluding the start and end of the utterance), and on the lexicon. This gives us nine retrieval measures for each experiment.

3.2 P_0 distribution

P_0 is the prior distribution over phonemes. We experiment with a uniform distribution ('uniform'), and one that is based on proportional counts in the corpus ('mle'). We evaluate using the log joint probability over time.

3.3 Gibbs sampling

The retrieved segmentation depends in part on the Gibbs sampling procedure. We experiment with different temperature regimes and initialisation strategies. Like the choice of P_0 distribution, we evaluate the temperature regimes and initialisation strategies using the log joint probability over time.

3.3.1 Temperature regime

We experimented with three different temperature regimes:

- Regime 0: 20000 iterations, from 0.1 to 1 in evenly spaced steps of 0.1
- Regime 1: 30000 iterations, from 0.1 to 1.5 in evenly spaced steps of 0.1
- Regime 2: 40000 iterations, from 0.002 to 1 in evenly spaced steps of 0.002

3.3.2 Initialisation

We experiment with initialisation with the true word boundaries as defined by the corpus as well as random boundary initialisation. In the random initialisation experiments, we first remove all boundaries and then randomly generated new boundaries. We experimented with different proportions of boundaries that were initialised with respect to the total number of possible boundaries (per utterance). We tested proportions 0, $\frac{1}{3}$, $\frac{2}{3}$, 1. The default was set to random initialisation with a fraction of $\frac{2}{3}$, as this was close to the true proportion in the corpus (29

3.4 Model parameters

The parameters of the DP model are α_0, which affects the number of word types proposed, and $p_{\#}$, the prior probability of a word boundary. We experiment with $\alpha_0 \in \{1, 2, 5, 10, 20, 50, 100, 200, 500\}$ and $p_{\#} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. During the α_0 experiments we kept $p_{\#}$ fixed to 0.5 and during the $p_{\#}$ experiments we kept α_0 fixed to 20.

3.5 PYP Model

Because h^- has to be modelled explicitly in the PYP model, the sampling is computationally a lot more expensive than the sampler used in DP sampling. Therefore a different iteration scheme was used with only 4000 iterations, and three equally long temperature steps (0.1, 1.1, 1.6).

3.5.1 Algorithm

Because of the different implementation of the Gibbs sampler in the PYP model, we first ran some control experiments, to confirm our implementation. The PYP model was run with the β parameter set to 0 and the results were compared to the DP model with the exact same parameters and temperature regime. Both models were expected to produce the same results.

Next, we examined the corpus probability across iterations, to confirm that our Gibbs sampler was reaching an optimum.

3.5.2 Parameters

Both the concentration parameter α and the smoothing parameter β affect the seating distribution. Because of this possible interaction, there might be no single best values for α and β. In order to find the best combination for both paramters, a grid search was conducted with $\alpha \in \{1, 2, 5, 10, 20, 50, 100, 500\}$ and $\beta \in \{0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 1, 0\}$

4 Results

4.1 P_0 distribution choice

Figure 1 shows the effect of phoneme distribution choice on the log joint probability. It turns out that the choice hardly influences the log joint probability at all.

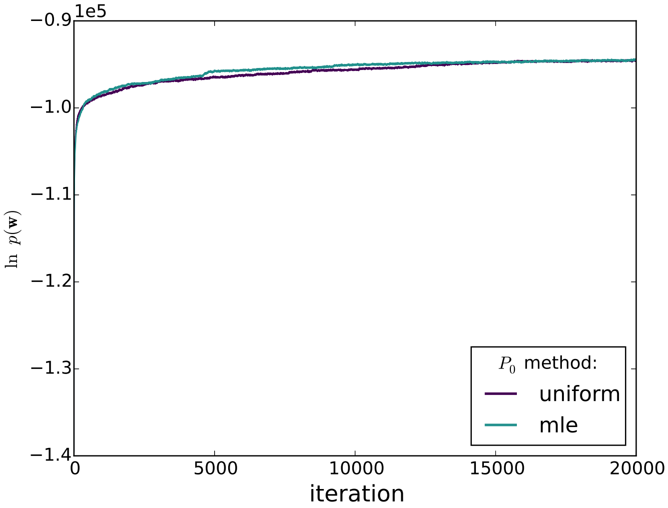


Figure 1: The effect of phoneme distribution choice for P_0 on the log joint probability over time.

4.2 Gibbs sampling

4.2.1 Temperature regime

Figure 2 shows the effect of temperature regime on the log joint probability over time. The temperature steps can clearly be seen by the sudden decreases in slope. We conclude that regime 2, which starts very low and increments the temperature by a small amount relatively often, leads to the highest log joint probability. However, for all regimes it seems that convergence is not reached in the end, as there is still some increase in probability.

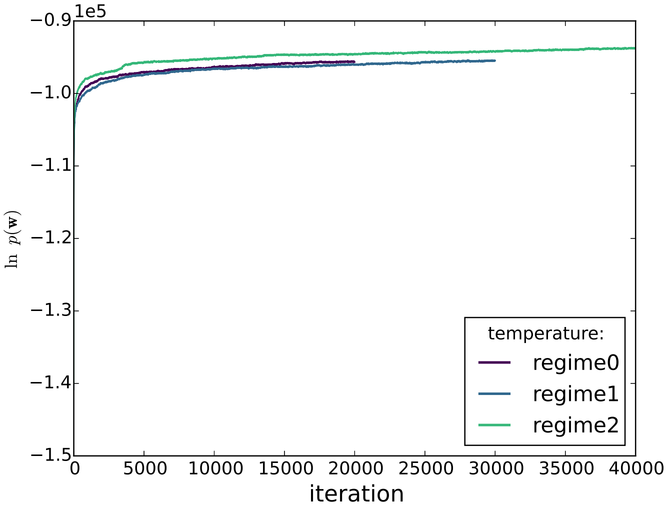


Figure 2: The effect of temperature regime on the log joint probability over time.

4.2.2 Initialisation strategy

Initialisation Figure 3 shows the effect of initialisation strategy on the log joint probability over time. As expected, the true initialisation results in the highest probability before the sampler is started. Running the sampler however does not increase the probability for this initialisation.

For all random initialisations, the sampler does increase the corpus probability, and seems to converge over time. Lower proportions of boundary initialisation result in higher probabilities.

Table 1: Retrieval quality of the model with $\alpha_0 = 20$ and $p_\# = 0.5$ evaluated on words, boundaries and lexicon.

	Precision	**Recall**	$\mathbf{F_0}$
Words	0.61	0.59	0.59
Boundaries	0.59	0.51	0.53
Lexicon	0.50	0.48	0.49

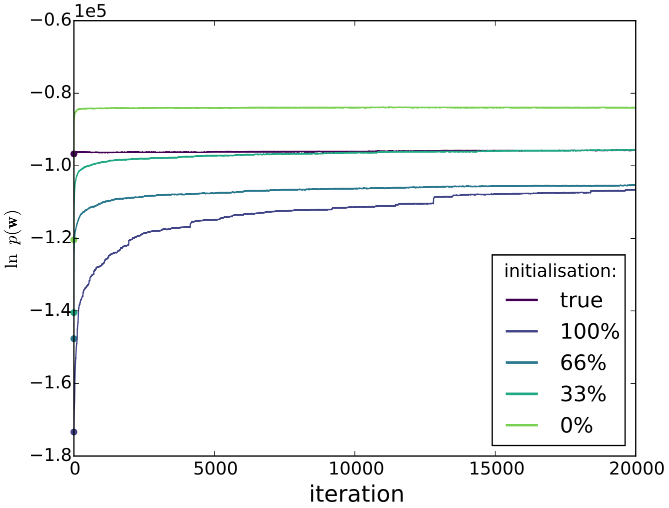
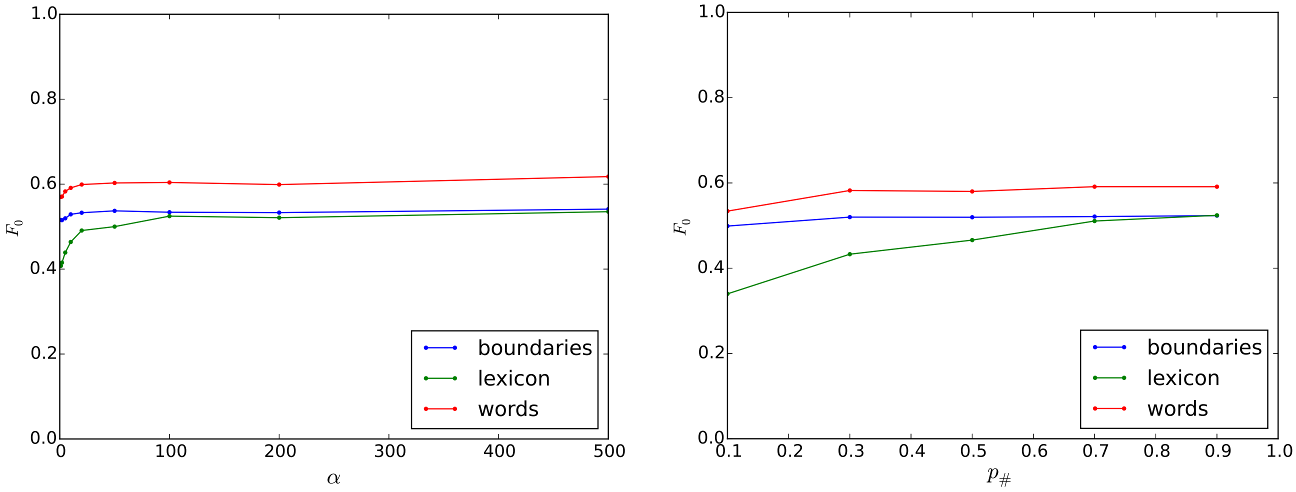


Figure 3: The effect of initialization strategy on the log joint probability over time. Percentage indicate the proportion of boundaries that are chosen (randomly). Dots indicate the value before sampling.

4.3 Model parameters

Figure 4 shows the effect of different model parameter choices on the F_0-measure, evaluated on words, boundaries and the lexicon. The effect is mostly visible at the lexicon level, favoring higher values for both α_0 and $p_\#$, but it does not seem to affect the F_0-measure of the words and boundaries much. Similar results (not reported) were found for precision and recall.



(a) F_0 for different values of α_0

(b) F_0 for different values of $p_\#$

Figure 4: The effect of different values for α_0 and $p_\#$ on retrieval quality evaluated on words, boundaries and the lexicon.

Table 1 shows the retrieval quality of the model evaluated on words, boundaries and lexicon.

4.4 Qualitative results

To find out why the model does not perform perfectly, we must analyze the segmentation that it produces. Below are some good and bad examples.

Good examples

- `fid It` (actual)
  `fid It` (retrieved)
- `pUt It In oke` (actual)
  `pUt It In oke` (retrieved)
- `oke` (actual)
  `oke` (retrieved)
- `gIv hIm 6 kIs oke kAm an` (actual)
  `gIv hIm 6 kIs oke kAm an` (retrieved)
- `lUk` (actual)
  `lUk` (retrieved)
- `D&t` (actual)
  `D&t` (retrieved)
- `WAt 6 n9s dOgi` (actual)
  `WAt 6 n9s dOgi` (retrieved)
- `lEts si nQ hu goz W* D&ts WAt 9d l9k tu no` (actual)
  `lEtssi nQ hu goz W* D&ts WAt 9d l9k tu no` (retrieved)

Bad examples

- `D&ts r9t` (actual)
  `D&tsr9t` (retrieved)
- `Ol r9t nQ WAt wUd yu l9k` (actual)
  `Olr9t nQWAt wUdyul9k` (retrieved)
- `WAt dId yu du t6de` (actual)
  `W AtdIdy udut6de` (retrieved)
- `WAt Els wUd yu l9k` (actual)
  `WAtEls wUdyul9k` (retrieved)
- `WAt Iz D&t` (actual)
  `WAtIzD&t` (retrieved)
- `wUd yu l9k D6 dOgi` (actual)
  `wUdyul9k D6dOgi` (retrieved)
- `k&n yu rid 6 bUk` (actual)
  `k&nyu ri d6bUk` (retrieved)
- `v*i gUd` (actual)
  `v*igUd` (retrieved)

We see that the good examples consist mostly of very short utterances, which are arguably easier to segment than longer utterances. The bad examples consist mostly of quite long utterances, and we can clearly see that they all lack some boundaries. The boundaries that the model did find in these cases, are mostly correct. In other words, the model under-segments these utterances.

4.5 PYP Model

4.5.1 Comparison to DP model

Figure 5 shows the comparison between the DP model and the PYP model with the β parameter set to zero. It can be seen that the models perform equally well. A investigation of the corpus probability over time showed that the convergence of both models was also similar. (see Figure 7)

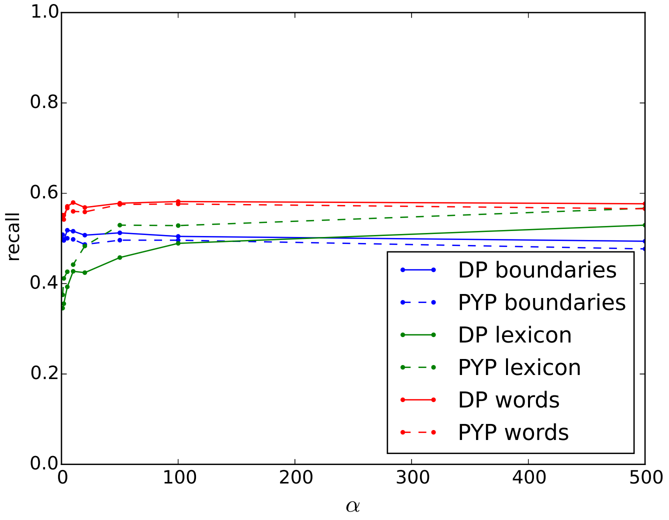


Figure 5: A comparison of the DP model with the PYP model (with β set to 0)

4.5.2 Parameter settings

Figure 6 shows the results of a grid search for different values of α and β. It can be seen that for each of the three measures, the best results are achieved when $\beta = 0$. When β is set to another value, performance decreases dramatically. Figure shows the comparison between the DP model and the PYP model with the β parameter set to zero. It can be seen that the models perform equally well. A investigation of the corpus probability over time showed that the convergence of both models was also similar.

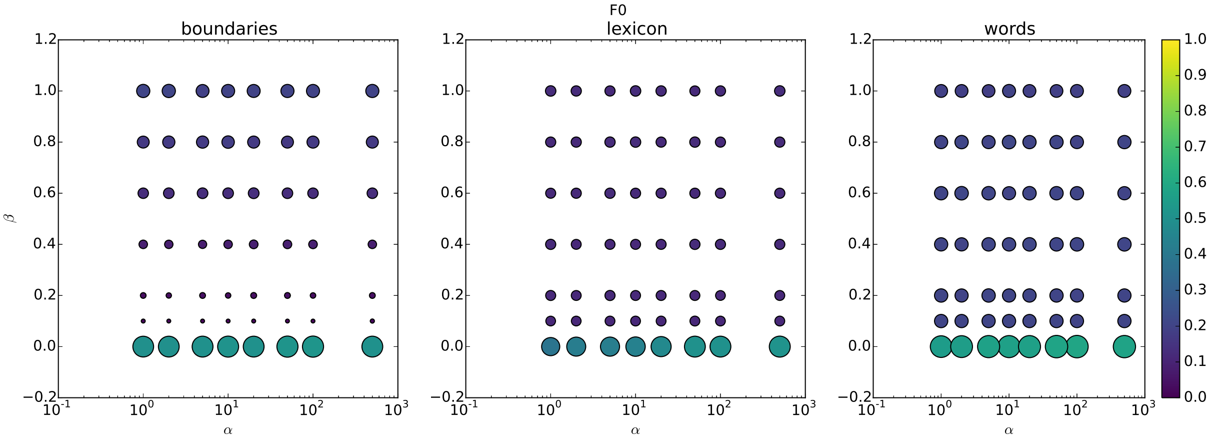


Figure 6: F_0 scores for different values for α and β using the PYP model.

4.5.3 Seating arrangement

Figure 7 shows the evolution of the seating arrangement over time. The most salient result is that the patterns for each measure are very different for $\beta = 0$ and all other values of β.

For all measures, we see a convergence for $\beta = 0$, to a value that is very different from the original value. For any other value of β, there is a small change in the first iteration, but from then on, all measures remain constant across iterations.

The number of types, the number of tokens and the number of tables (K) decreases as the iterations progress. All these measures increase for higher values of β.

In general $\beta = 0$ commits to this trend, except for the number of tokens, which is high compared to other values of β.

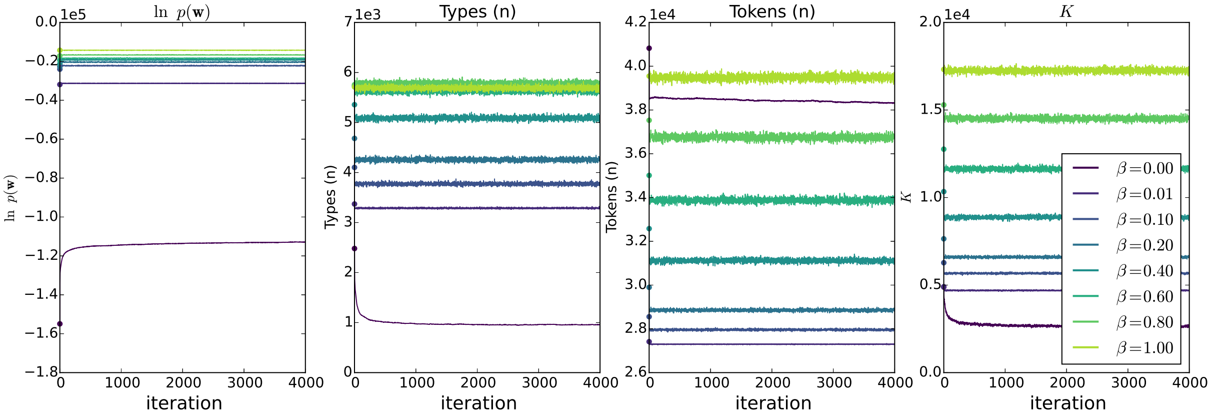


Figure 7: The evolution of various seating arrangement measures over time, for different values of β, α=2 for all graphs.

4.5.4 Qualitative analysis

Below we list some of the errors that our model made. Under segmentation does occur, but in contrast with the DP model, we find many incorrect segmentations that are 'just wrong'. Some words are under segmented, whereas others are over segmented within the same sentence.

- D6 dOgi l9ks tu ste In D* (actual)
  D6dOgil9 kstust e InD* (retrieved)
- si (actual)
  s i (retrieved)
- pUt D6 dOgi In hIz hQs (actual)
  pU tD6d Og iI nh IzhQs (retrieved)
- go an (actual)
  goa n (retrieved)
- pUt hIm In hIz hQs (actual)
  pU thImInhIzhQs (retrieved)
- v*i gUd (actual)
  v* i g Ud (retrieved)
- nQ hi wants Qt 6gEn (actual)
  n Q h iwa n tsQt 6 gEn (retrieved) (retrieved)

5 Discussion

Of all the experiments, the initialisation strategy shows the largest effect on performance. Although initialising the corpus with the true segmentation results in the highest probability before sampling, a higher probability is reached using other initialisations. The fact that there are higher probabilities than the true segmentation does indicate a fundamental limitation of our model.

A worrying finding is the fact that initialisation has a large effect on the corpus probability after running the sampler. This indicates that our sampler does not converge to a global optimum but get's stuck locally. Furthermore, this finding is not in line with [2] who found no effect of initialisation.

Moreover, we note that although the initialization with true boundaries has the highest joint probability on the corpus initially, it does not get the highest final joint probability. Assuming the true segmentation of the corpus is correct, we conclude that the model is in fact inadequate for this problem. This suggests that the assumption that words are independent units may not be a realistic assumption, and should be dropped or changed (to e.g. the bigram assumption).

Interestingly, the probability of the true initialisation does not change after sampling, but remains constant indicating that the true segmentation is in fact a local optimum. Furthermore, fewer boundaries result in a higher probability with the best probability found when no boundaries are initialised. This initialisation corresponds to the trival MLE segmentation where each utterance is seen as a token.

Varying the parameters only has marginal effects. The temperature regime can slightly improve learning speed and performance, but it does not make a big difference. The α_0 parameter should not be too small, but after a certain value (e.g. $\alpha_0 > 20$), it does not really affect retrieval quality anymore. The same applies to the $p_\#$ parameter, where values greater than 0.5 are preferable. Finally, the fact that the lines in most graphs don't seem to converge, suggests that the model may be stuck in a local optimum.

The numeric results of our experiments do not directly match those of [2], but they are similar. We suspect that these differences may be caused by small implementation differences. However, our model suffers from the same problem as their unigram model: it tends to under-segment the corpus. Thus, our results confirm an important finding of [2], which is that the assumption that words are independent of each other is most likely inadequate.

Taken altogether, the DP unigram model does not perform very well on this task, in line with the findings of [2]. We suggested the PYP as an improvement to this model, but in fact, the PYP model worsened performance. Either the model and parameters were not implemented correctly, or the PYP model is actually not suited for this task. We conducted several experiments to investigate both options.

With β set to 0, the model behaves as the DP model, as expected. This indicates that the of modelling h^- is implemented correctly. Another expected finding is that higher β values result in an increased number of clusters (K).

However we also found some unexpected effects of the β parameter. First of al, lower the values of β results in decreased performance. As the highest performance was found for $\beta = 0$, this finding is counter intuitive, and may indicate an implementation error.

As expected, larger β values result in more clusters. The number of types also increases with β, which means that not only do we have more clusters per type, but there also are more different labels. Surprisingly the number of tokens increases with β as well. Taken together with the qualitative results, we conclude that the model does still under-segment but also makes many incorrect segmentations.

The different values of β also have a large effect on corpus probability, even after initialisation. This may mean indicate an implementation error and may be due to the fact that we defined the corpus probability as $p(\mathbf{w}|\mathbf{z})$ instead of including the distribution over $\mathbf{z}$. However the same probability is used in [2] for the bigram model. Analogous to our PYP model, the seating is modelled explicitly for the bigram model. The probability of [2] does converge nicely, which means that our equations for $p(w_i|\mathbf{w}_{-i}, \mathbf{z}^-)$ may be incorrect.

Besides possible implementation errors, our finding may imply that the PYP model is not suited for this task. It could be the case that the distribution of types and tokens in infant directed language is not distributed according to the power-law. In fact, this may very well be the case, as the vocabulary of our corpus is rather limited. Furthermore, our findings may indicate that the assumption of the distribution over words that is implied by the PYP is not an assumption that an ideal learner should make in order to find the right clustering. However, more sanity checks and control experiments have to be conducted to find out what exactly is causing our results.

6 Conclusion

In this project, we reproduced the unigram model of [2] and attempted to replicate their results. While we were unable to exactly match their results, our results generally seem to agree with their results. Moreover, our results show that our model suffers from the same problem as their unigram model, namely that it tends to under-segment the corpus. This confirms their main finding that the word-to-word independence assumption is inadequate and should be dropped or changed. Extending the DP unigram model to a PYP model did decrease the performance. Further research should investigate what causes this effect.

References

[1] Richard N Aslin, Jenny R Saffran, and Elissa L Newport. Computation of conditional probability statistics by 8-month-old infants. *Psychological science*, 9(4):321–324, 1998.

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

[3] Sharon Goldwater, Mark Johnson, and Thomas L Griffiths. Interpolating between types and tokens by estimating power-law generators. In *Advances in neural information processing systems*, pages 459–466, 2005.

[4] Zellig S Harris. *From phoneme to morpheme.* Springer, 1970.

[5] Jim Pitman and Marc Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.

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

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