Long-tail Distributions and Unsupervised Learning of Morphology

Qiuye Zhao¹ Mitch Marcus¹ (1) University Of Pennsylvania qiuye@cis.upenn.edu, mitch@cis.upenn.edu

Abstract

In previous work on unsupervised learning of morphology, the long-tail pattern in the rank-frequency distribution of words, as well as of morphological units, is usually considered as following Zipf's law (power-law). We argue that these long-tail distributions can also be considered as lognormal. Since we know the conjugate prior distribution for a lognormal likelihood, we propose to generate morphology data from lognormal distributions. When the performance is evaluated by a token-based criterion, giving more weights to the results of frequent words, the proposed model preforms significantly better than other models in discussion. Moreover, we capture the statistical properties of morphological units with a Bayesian approach, other than a rule-based approach as studied in (Chan, 2008) and (Zhao and Marcus, 2011). Given the multiplicative property of lognormal distributions, we can directly capture the long-tail distribution of word frequency, without the need of an additional generative process as studied in (Goldwater et al., 2006).

Keywords: Morphological Learning, Zipf's law, Lognormal distribution, Long tail distribution, Gibbs Sampling, Bayesian approach.

1 Introduction

Unsupervised learning of morphology is an active research area. In this work, we will focus on learning segmentations of words¹. A segmentation of word w can be denoted as w = t.f, which means that concatenating stem t and suffix f gives word w.

Assuming that stems and suffixes are independently distributed, a baseline generative morphology model is like this,

 k^{T}, k^{F} = number of distinct stem types, number of distinct suffix types α^{T}, α^{F} = constant hyper-parameters $\theta^{T}, \theta^{F} \sim$ Symmetric-Dirichlet_{k^T}(α^{T}), Symmetric-Dirichlet_{k^F}(α^{F}) $t_{i=1...N}, f_{i=1...N} \sim$ Multinomial(θ^{T}), Multinomial(θ^{F}) $w_{i=1...N} \sim I(w = t.f)P(t|\theta^{T})P(f|\theta^{F})$

where N is the number of words and I(w = t.f) is the indicator function taking on value 1 when concatenating stem t and suffix f gives word w and 0 otherwise.



Figure 1: Observing long-tail distributions of word frequency in WSJ Penn Treebank.

In the above baseline model, there no special statistical property of word frequency is captured, nor of morphological units (e.g. stems or suffixes). As depicted in Figure 1-a, it is well-known that given a large corpus, a rank-frequency plot of words generally exhibits a long tail that is distinctively heavier than in normal (Gaussian) distributions. An alternative way to show this long-tail pattern is by plotting the corresponding Complementary Cumulative Distribution Function (CCDF) on logistic scales. As depicted in Figure 1-c, the CCDF plot behaves like a straight line and is smoother than the corresponding log-log rank-frequency plot in Figure 1-b, which also approaches a straight line for long-tail distributions. When words are segmented into stems and suffixes, distributions of these units can also be examined.

As we are going to show in Section 2.3, the rank-frequency distributions of morphological units also show up as straight lines in log-log rank-frequency plots and CCDF plots. In previous work on morphological learning, such as (Chan, 2008), (Zhao and Marcus, 2011) and (Goldwater et al., 2006, 2011), the straight lines on logistic scales are usually interpreted as following Zipf's law (Zipf, 1949). On the other hand, lognormal distributions with large variance also yield straight lines on both the log-log rank-frequency plot and the CCDF plot. In Section 2, we are going to discuss

¹ A segmentation model may not be well-defined for morphology, especially when morph-rich languages are considered; however, different forms of morphological analyses may be compared as simple segmentation analyses.

more about the long-tail distributions and argue that the signature straight lines may suggest either power-law or lognormal.

So as to take advantage of the special statistical property of morphological units, which is considered as following the Zipf's law in (Chan, 2008), Chan proposes a rule-based bootstrapping algorithm for morphology learning, which is revised in (Zhao and Marcus, 2011) for acquiring functional elements. Even though Zipf's law is discussed in both works, its specific definition does not really matter in the design of the algorithms. For the sake of comparison with other models, we implement a reduced version of the bootstrapping algorithm as described in (Zhao and Marcus, 2011), eliminating all ad-hoc linguistic assumptions encoded in the original algorithm. With this rule-based algorithm, the acquired segmentation model performs rather well when evaluated with the type-based criterion, but notably bad with the token-based evaluation, which gives more weights to the results of frequent words than the type-based evaluation. We will describe this algorithm in Section 3.1 with more details.

The rule-based bootstrapping algorithm utilizes type frequencies only, no matter what form of input is given. On the other hand, as shown in (Goldwater et al., 2006), when Dirichlet-multinomial model is assumed, the option of utilizing token frequency in generative model doesn't help, and in the contrast it hurts the inference of the generative model. Goldwater et al. (2006) argued that this morphology model doesn't capture the special statistical property of word frequency, therefore, an addition generative process is introduced to transform the word frequencies to exhibit the desired distribution. Since the long-tail pattern in word frequency is considered as following Zipf's law, a generalized Chinese restaurant process, Pitman-Yor process (Pitman and Yor, 1997), is exploited for producing power-law distributions. For the sake of comparison, we re-implement the morphology model in (Goldwater et al., 2006), and conduct more experiments with different configurations.

We propose to compute lognormal likelihood, instead of multinomial likelihood, for both stems and suffixes, in the generative morphology model. With this lognormal model, the option of utilizing token frequency, i.e. taking input of the unprocessed text data, dose help the inference of the generative model, especially when the token-based evaluation is preformed. When evaluated by the token-based criterion, which gives more weights to the results of frequent words, the proposed model performs significantly better than other models in discussion, no matter what form of input is fed to the multinomial model or the rule-based algorithm.

With the proposed model, the particular statistical properties of stems and suffixes are utilized in a Bayesian model instead of a rule-based model. Furthermore, as we will discuss in Section 2.3, given the multiplicative property of lognormal distributions, the word frequency distribution can also be predicted as lognormal. Therefore, we can directly capture the statistical property of word frequencies without the need of an additional generative process. Especially, the proposed generative model is more accurate with the token-based evaluation when utilizing token frequency, and more accurate with the type-based evaluation when utilizing type-frequency. This result pattern suggests that the proposed model is able to adapt to real data distribution by itself, therefore, we do not need to concern with justifying the appearance of type frequencies in morphology learning, as pursued in (Goldwater et al., 2006).

We are going to use Gibbs sampling, a standard Markov Chain Monte Carlo (MCMC) method for the inference of generative models. In each iteration, the morphological analysis of each word is sequentially re-sampled from its conditional posterior given morphological analyses of all other words. Since the sampling process is much more complex and time-consuming for the lognormal model than the multinomial model, we propose to constrain the learning of generative models with the acquisition outputs of the rule-based model. Since the rule-based bootstrapping algorithm takes input of raw corpora only and so the generative models, the combination of these two processes results in a totally unsupervised learning process as well. Even though this method is motivated by the concern of training efficiency, the proposed use of acquisition outputs from a rule-based model also significantly improves the performance of generative models, consistently for both the lognormal and the multinomial model.

2 Long-tail distributions

2.1 The long-tail pattern

Given a large corpus, we can compute the word frequency of each word type by counting its occurrences in the corpus. When we plot word frequency against its rank, such as in Figure 1-a, there is a long tail of the curve composed of the large number of words that occur in low frequency. When plotted on a logistic scale, as in Figure 1-b, the rank-frequency plot behaves like a straight line. An equivalent form of the rank-frequency approach is to plot the corresponding Complementary Cumulative Distribution Function (CCDF). Instead of plotting a function of rank, we can also plot $P(F \ge f)$ as a function of frequency f. As shown in Figure 1-c, the CCDF plot also behaves like a straight line on the logistic scale, which is smoother than the log-log rank-frequency plot.

We generally refer this kind of distributions as long-tail distributions, observing that a large portion of its population are composed of low-frequency events, which form a longer tail than normal (Gaussian) distributions. Long-tail patterns have been widely observed in various fields, but they may be studied as different distributions. For example, economists may be familiar with this pattern as Pareto distribution, which is also known as '80-20' rule. In Pareto's original study (Pareto, 1896), the long-tail pattern is shown on CCDF plots, so it took a while for people to understand that it is a power-law distribution and is synonymous with 'Zipf's law' (Newman, 2005). Zipf's law is proposed by linguist George Kingsley Zipf in his study of vocabulary distribution, and is widely used to interpret the straight lines on logistic scales in the study of language. In more recent works, the convention of treating long-tail patterns as power-law distributions has been challenged. For example, Downey (Downey, 2001) argues that many networks metrics, such as file sizes and transfer times, should be modeled as lognormal distributions. Lognormal distributions with large variance also yield straight lines on the log-log rank-frequency plot and the CCDF plot.

2.2 Generating Zipf's law and lognormal distributions

Based on the idea of preferential attachment, i.e. a 'richer-get-richer' process, if we generate new word occurrences more likely of popular word types than of rarely seen word types in previous process, then word frequency of the generated corpus may follow Zipf's law or be lognormal, depending on subtle differences in the generative processes.

More specifically, suppose that we are given *i* words for a start, $i \ge 1$. Let n_k^i denote the number of occurrences of all the words that occur exactly *k* times in the previous *i* words. Let $P(w_{i+1} = k)$ denote the probability that the $i + 1_{th}$ occurrence is a word that has already appeared *k* times in the previous *i* words. Consider the following process as described in (Simon, 1955),

$$P(w_{i+1}=k)=\alpha n_0+F_i n_k^i,$$

where n_0 and α are constants. If $F_i = \frac{(1-\alpha)}{i}$, then asymptotically, $P(w_i = k)$ will approach a power-law distribution. On the other hand, if the constant item is removed from the above process,

$$P(w_{i+1} = k) = F_i n_k^i$$

and F_i are independent and identically distributed variables with finite mean and variance, then asymptotically, $P(w_i = k)$ will approach a lognormal distribution.

A even more naive generative model for Zipf's law is Miller's monkey (Miller, 1957), who can not only type with a keyboard, but also distinguish space bar from other keys. If Miller's monkey manages to hit the space bar with a constant probability and never hits the space bar twice subsequently, then the word frequency in the monkey's output follows a power law. One crucial assumption in Miller's demonstration is that all non-space letters are hit with equal probabilities. However, for the case that any two letters are hit with different probabilities, Perline (1996) argues that for all words of length up to a constant, their rank-frequency distribution converges to a lognormal distribution.

After reviewing a brief history of generative models for power-law and lognormal distributions, Mitzenmacher (2004) suggests that "It might be reasonable to use which ever distribution makes it easier to obtain results."²



Figure 2: Observing long-tail distributions for morphological units.

2.3 Lognormal distributions

One advantage of modeling long-tail patterns as lognormal is so that the multiplicative property holds, i.e. the product of independent lognormally-distributed random variables is itself lognormally distributed. For example, consider the baseline generative morphology model as described in the introduction, P(w) = P(t)P(f). If both stems and suffixes are lognormally distributed, then word frequency is also lognormally distributed.

So as to examine the distributions of morphological units, with the help of the gold part-of-speech annotations, we segment all verbs into stems and suffixes in WSJ Penn Treebank (Marcus et al., 1993). As shown in Figure 2-a and b, the signature straight lines on CCDF plots suggest that both stems and suffixes of verbs can be modeled as lognormal. Then, given the multiplicative property of lognormal distributions, verbs should also be lognormally distributed, which is confirmed by its CCDF plot in Figure 2-c.

Another reason for one to consider modeling long-tail patterns as lognormal distributions is that we know the conjugate prior distribution for a lognormal likelihood, but not for power-law likelihood. Considering the generative model P(w) = P(t)P(f), if we assume that *t* and *f* are lognormal, their mean μ and variance σ^2 can be drawn from Normal priors and Inverse-Gamma priors respectively.

² As also pointed out in (Mitzenmacher, 2004), if a power law distribution can have infinite mean and variance, then it is inaccurate to analyze it as lognormal. In present examples, we assume the exponent of power-law distributions is greater than 0, thus it is safe for us to experiment with either distribution.

3 A rule-based model

Given a set of stems \mathbb{T} and a set of suffixes \mathbb{F} , we can divide a word *w* into stem *t* and suffix *f*, if $t \in \mathbb{T}$, $f \in \mathbb{F}$ and w = t.f. For example, if $\mathbb{T} = \{$ *'laugh-'*, *'analyz-'* $\}$, and $\mathbb{F} = \{$ *'-ed'*, *'-s'* $\}$, then *'analyzed'* can be segmented into '*analyz-'* and '*-ed'*, but '*red'* won't be segmented. So as to learn such a rule-based morphology model, we need to acquire a set of stems and a set of suffixes.

3.1 A bootstrapping algorithm for acquiring morph units

In this section, we are going to describe a bootstrapping algorithm, adapted from the algorithm for acquiring functional elements in (Zhao and Marcus, 2011). This line of algorithms is especially designed to account for the long-tail pattern observed for stems and suffixes, but not specific to either Zipf's law or lognormal distributions. As stated in (Chan, 2008), in which a bootstrapping algorithm is originally proposed for acquiring transformation-based morphological rules, "what matters, though, is that the number of types per inflection decreases rapidly, ..., there are few highly frequent inflections, and many more infrequent ones."

Following (Zhao and Marcus, 2011), our algorithm is built upon a distributional property of 'functional elements': they occur in diverse contexts. In the case of learning morphological segmentations, 'functional elements' are suffixes in context of stems. For example, an inflectional ending '-ed' can be concatenated to most verb stems to derive past tense forms, in contrast to which a non-sense suffix '-roached' can only be seen in few particular word types.

If all prefixes in all possible divisions of a word count as stems, then as computed from the WSJ corpus, the top three suffixes with the highest contextual diversity will be '-s', '-d' and '-e', two of which do not comply with common sense of morphological suffixes in English. In other words, for acquiring suffixes, we want to compute their contextual diversity according to properly justified stems only. The most simple way of justifying stems as proper contexts for suffixes is to check whether it serve as context of more than one type of suffixes. For example, stem '*laughin-*' should not be justified, because except for a particular suffix '-g', it cannot be concatenated with other suffixes to form legal words.

Given a set of properly justified stems T, we measure the contextual diversity of a suffix f as

$$div(f,\mathbb{T}) = \sum_{t\in\mathbb{T}} \delta(t.f),$$

where $\delta(t.f)$ is set to 1 if *t*.*f* forms any word, otherwise 0. For example, if we are given a set of properly justified stems, including '*laugh-*' but not '*b-*', the diversity measurement of '*-ing*' will increase by one given the existence of word '*laughing*' but not word '*bing*'.

Algorithm 1 The bootstrapping algorithm for acquiring stems and suffixes
Require: A corpus C containing raw text only.
Initialize set \mathbb{F}_0 to be empty and set \mathbb{T}_0 to contain all possible prefixes
for $k = 1K$ iterations do
Let \mathbb{F}_k contain the top k suffixes with the highest diversities measured by $div(f, \mathbb{T}_{k-1})$.
Let \mathbb{T}_k contain stems that form legal words with suffixes in \mathbb{F}_k
end for
return \mathbb{F}_{ν} and \mathbb{T}_{ν}

We implement a reduced version of the bootstrapping algorithm in (Zhao and Marcus, 2011),

eliminating all ad-hoc linguistic assumptions encoded in the original algorithm. As depicted in Algorithm 1, the algorithm generates two sets of acquisition outputs during the bootstrapping process, both of which justify the proper set for measuring contextual diversity for each other. As the two sets alternately update during the bootstrapping process, the diversity measurement of either set is expected to be more and more accurate.

The only required input to this algorithm is a corpus \mathbb{C} of raw text without any form of annotation. Set \mathbb{F}_0 is initialized to be empty and set \mathbb{T}_0 is initialized to contain all prefixes in all possible divisions of all words in corpus \mathbb{C} . At the k_{th} bootstrapping iteration, k > 0, we compute set \mathbb{F}_k as the top k suffixes of the highest contextual diversity according to set \mathbb{T}_{k-1} . And set \mathbb{T}_k contain stems that can form legal words with suffixes in \mathbb{F}_k . Since the diversity measurement of suffixes varies over iterations with respect to updated set of stems, a suffix that is selected to output at some iteration, is not guaranteed to be selected in the following iterations.

kth iter.	set ⊮	size of set \mathbb{T}			
1th	-\$	668			
5th	-ed, -ing, -s, -e, -es	2274			
10th	-ed, -ing, -e, -s, -es, -er, -ers, -ion, -ions, -ly	2776			
20th	, -ion, -ers, -y, -ions, -al, -or, -ors, -ings, -able, -ive, -ly, -aly, -ies	2993			
Fix set \mathbb{T} with all prefixes in all possible divisions of words.					
k=20	-s, -d, -e, -ed, -g, -n, -ng, -y, -ing, -t, -r, -es, -er, -on, -l, -rs, -a, -ly, -ion, -0	7091			

Table 1: The acquisition outputs by Algorithm 1 over WSJ Penn Treebank.

We run this bootstrapping algorithm for acquiring stems and suffixes from the WSJ corpus. For the sake of comparison, we also experiment without updating set \mathbb{T} during the bootstrapping. With the unchanged set \mathbb{T} , which is initialized to contain all prefixes in all possible divisions of all words in corpus \mathbb{C} , the bootstrapping algorithm is degenerated to a simple counting function, which, for a given *k*, returns the top *k* frequent suffixes. So as to compare with other models more fairly, we didn't implement the mechanisms in the original algorithm for removing complex suffixes such as *-ers*, *-ings* and *-ors*, neither the trick for removing the most noisy suffix *-e*.

4 A generative model with multinomial likelihood

In this section, we are going to describe a generative morphology model that involves one more random variable than the baseline model we sketched in the introduction. In the baseline generative model, we assume without any condition that both stems and suffixes are independently and multinomially distributed. For the current model, stems and suffixes are independently and multinomially distributed in each inflectional class. A morphological analysis of word w can be denoted as (c, t, f), which means that $w = t \cdot f$ and this analysis belongs to inflectional class c.

Assume a multinomial distribution over k^c inflectional classes, with parameters θ^c . To make predictions about new classes, we take symmetric Dirichlet priors α^c on parameters θ^c , which means that the way each inflectional class is used has little variation. When k^c is set as 1 in this model, it degenerates to the baseline generative model that assumes stems and suffixes are independently distributed. Again, let *N* be the number of words and I(w = t.f) denote the indicator function taking on value 1 when concatenating stem *t* and suffix *f* gives word *w* and 0 otherwise. Our generative morphology model is like this,

$$\begin{split} k^{C} &= \text{number of inflectional classes} \\ \alpha^{C}, \alpha^{T}, \alpha^{F} &= \text{constant hyper-parameters} \\ \theta^{C} &\sim \text{Symmetric-Dirichlet}(\alpha^{C}) \\ \theta^{T}_{i=1...k^{C}}, \theta^{F}_{i=1...k^{C}} &\sim \text{Symmetric-Dirichlet}(\alpha^{T}), \text{Symmetric-Dirichlet}(\alpha^{F}) \\ c_{i=1...N} &\sim \text{Multinomial}(\theta^{C}) \\ t_{i=1...N}, f_{i=1...N} &\sim \text{Multinomial}(\theta^{T}_{c_{i}}), \text{Multinomial}(\theta^{F}_{c_{i}}) \\ w_{i=1...N} &\sim I(w_{i} = t.f)P(c_{i} = c|\theta^{C})P(t|c,\theta^{T})P(f|c,\theta^{F}) \end{split}$$

This is the morphology model of choice in (Goldwater et al., 2011), following which we also use Gibbs sampler, a simple and widely-used Markov Chain Monte Carlo method, for inference. Assume the exchangeability of morphological analyses. The finite set of morphological analyses $\{a_1, ..., a_N\}$ is exchangeable, if for a permutation, π , of the integers from 1 to *N*,

$$P(a_1,...,a_N) = P(a_{\pi(1)},...,a_{\pi(N)}).$$

At each iteration, from $\mathbf{a} = \{a_1, ..., a_N\}$, sample a'_1 given morphological analyses of all other words, i.e. $A_{-1} = \{a_2, ..., a_N\}$, then go to $\{a'_1, a'_2, ..., a_N\}$ and so on until $\{a'_1, a'_2, ..., a'_N\} = \mathbf{a}'$. It can be shown that this sampling process defines a Markov chain on $\mathbf{a}, \mathbf{a}', \mathbf{a}'',$ After a sufficient amount of time, the probability values are independent of the starting values and tend towards the stationary distribution $P(\mathbf{a})$. More about Gibbs sampling will be discussed in Section 5.2, where sampling processes for two generative models are compared.

Related work

The Dirichlet-multinomial model is not able to capture neither the particular statistical property of word frequencies, as studied in (Goldwater et al., 2006), nor the particular statistical property of morphological units, as studied in (Chan, 2008) and (Zhao and Marcus, 2011). As described in Section 3.1, a rule-based bootstrapping algorithm can be designed to take advantage of long-tail distributions of both stems and suffixes. However, as we will show with experimental results in Section 6, the rule-based model performs notably bad with the token-based evaluation, which gives more weights to the results of frequent words than the type-based evaluation.

In a two-stage learning framework proposed in (Goldwater et al., 2006), the morphology model as introduced above is used as a 'generator' for producing words. Word frequencies are then transformed to exhibit power-law by an additional generative process, called 'adaptor'. Especially, a two-parameter generalization of Chinese Restaurant Process (Pitman and Yor, 1997) is used as an adaptor. The Pitman-Yor process implements the principle of preferential attachment and guarantees the exchangability of its outputs. As we will also show in Section 6, the generator, i.e. the Dirichlet-multinomial model, learns reasonably well from the input of distinct word types, i.e. type-based input; however, the multinomial model itself cannot adapt to the input of unprocessed text data, i.e. the token-based input. Augmented with an adaptor, the generator may achieve its best performance with all forms of input; however, the introduction of such an adaptor does not improve the overall performance of the generator. In the following section, we will propose a generative model that learns well from both forms of input: type-based or token-based, without the need of an adaptor. The proposed model is able to utilize token frequency by itself, therefore, we do not need

to concern with justifying the appearance of type frequency in morphology learning, as pursued in (Goldwater et al., 2006).

5 A generative model with lognormal likelihood

We propose to compute lognormal likelihood, instead of multinomial likelihood, for both stems t and suffixes f. In this way, the particular statistical property of stems and suffixes is utilized in a Bayesian model other than a rule-based model. Furthermore, as we discussed in Section 2, given the multiplicative property of lognormal distributions, for each inflectional class, the generated word distribution is also lognormal. Therefore, with the proposed model, we can directly capture the statistical property of word frequencies without the need of an additional generative process.

5.1 The probability model

Again, assume that stems and suffixes are independent given inflectional classes. We still have a multinomial distribution over k^c classes, with parameters θ^c , and take symmetric Dirichlet priors α^c on θ^c . Now, in each inflectional class c, we assume a lognormal distribution of frequency for both stems and suffixes. It is equivalent to assume that the logarithms of stem/suffix frequency are normally distributed over the rank. For example, if the logarithms of stem frequency, $\mathfrak{L}(t)$, is normally distributed with mean μ^T and variance $(\sigma^T)^2$, then stem frequency t is lognormally distributed with mean e^{μ^T} and variance $(\sigma^T)^2$.

For a random variable *X* that is normally distributed, if both its mean μ and variance σ^2 are random, we will use the following distribution for priors, which can be shown to be conjugate to normal likelihood. Assume μ_0, γ_0, α , and β as the constant hyper-parameters, then we have

$$\sigma^2 \sim \text{Inverse-Gamma}(\alpha, \beta)$$

 $\mu | \sigma^2 \sim \text{Normal}(\mu_0, \gamma_0 / \sigma^2)$
 $x | \mu, \sigma^2 \sim \text{Normal}(\mu, \sigma^2).$

In our case, we construct the probability distributions as follows:

$$\theta^{C} \sim \text{Dirichlet}(\alpha^{C})$$

$$c_{i=1...N} \sim \text{Multinomial}(\theta^{C})$$

$$(\sigma_{i=1...k^{C}}^{T})^{2}, (\sigma_{i=1...k^{C}}^{F})^{2} \sim \text{Inverse-Gamma}(\alpha^{T}, \beta^{T}), \text{Inverse-Gamma}(\alpha^{F}, \beta^{F})$$

$$\mu_{i=1...k^{C}}^{T}, \mu_{i=1...k^{C}}^{F} \sim \text{Normal}(\mu_{0}^{T}, \gamma_{0}^{T}/(\sigma_{i}^{T})^{2}), \text{Normal}(\mu_{0}^{F}, \gamma_{0}^{F}/(\sigma_{i}^{F})^{2})$$

$$t_{i=1...N}, f_{i=1...N} \sim \text{Log-Normal}(e^{\mu_{c_{i}}^{T}}, (\sigma_{c_{i}}^{T})^{2}), \text{Log-Normal}(e^{\mu_{c_{i}}^{F}}, (\sigma_{c_{i}}^{F})^{2})$$

$$w_{i=1...N} \sim I(w_{i} = t.f)P(t|c_{i}, \mu_{c_{i}}^{T}, (\sigma_{c_{i}}^{T})^{2})P(f|c_{i}, \mu_{c_{i}}^{F}, (\sigma_{c_{i}}^{F})^{2})$$

where $\alpha^C, \mu_0^T, \gamma_0^T, \alpha^T, \beta^T, \mu_0^F, \gamma_0^F, \alpha^F$ and β^F are constant hyper-parameters .

5.2 Gibbs Sampling

As discussed in Sect 4, we use Gibbs sampler for the inference of generative models. At each iteration, we need to sample a morphological analysis for each word given the morphological analyses of all other words. For example, at initialization, each word receives a random analysis, then we proceed by sampling a morphological analysis a'_1 of the first word, w_1 , given the random

Multinomial	$(c,t,f) w_i,\mathbf{A}_{-i} \sim I(w_i = t.f) \frac{a^c + m_e}{k^c a^c + N} \frac{a^T + m_{e,t}}{k^T a^T + m_e} \frac{a^F + m_{e,f}}{k^F a^F + m_e}$			
	$c \mathbf{A}_{-i}\sim rac{lpha^{C}+m_{c}}{k^{C}lpha^{C}+m_{c}}$			
	$(\sigma_c^T)^2 \mathbf{A}_{-i} \sim IG(\alpha^T + \frac{m_{c,t}}{2}, \beta^T + \frac{\sum (\mathfrak{L}(t_i) - \overline{\mathfrak{L}(t)})^2}{2} + \frac{m_{c,t} \gamma_0^T (\overline{\mathfrak{L}(t)} - \mu_0^T)^2}{2(\gamma_0^T + m_{c,t})})$			
	$\mu_c^T (\sigma_c^T)^2, \mathbf{A}_{-i} \sim N(\frac{m_{c,t} \overline{\mathcal{L}(t)} + \gamma_0^T \mu_0}{m_{c,t} + \gamma_0^T}, \frac{\sigma_c^T}{m_{c,t} + \gamma_0^T})$			
	$(\sigma_c^F)^2 \mathbf{A}_{-i} \sim IG(\alpha^F + \frac{m_{c,f}}{2}, \beta^F + \frac{\sum (\mathfrak{L}(f_i) - \overline{\mathfrak{L}(f)})^2}{2} + \frac{m_{c,f} \gamma_0^F (\overline{\mathfrak{L}(f)} - \mu_0^F)^2}{2(\gamma_0^F + m_{c,f})})$			
	$\mu_c^F (\sigma_c^F)^2, \mathbf{A}_{-i} \sim N(\frac{m_{c,f}\overline{\mathcal{Q}(f)} + \gamma_0^F \mu_0}{m_{c,f} + \gamma_0^F}, \frac{\sigma_c^F}{m_{c,f} + \gamma_0^F})$			
Lognormal	$(t,f) w_i, \mathbf{A}_{-i} \sim I(w_i = t,f) \texttt{lognormal}(e^{\mu_c^T}, (\sigma_c^T)^2) \texttt{lognormal}(e^{\mu_c^F}, (\sigma_c^F)^2).$			

Table 2: Sample a morphological analysis of the word w_i .

analyses of all other words, $A_{-1} = \{a_2, a_3, ..., a_N\}$. Then we sample a morphological analysis a'_2 of word w_2 , given $A_{-2} = \{a'_1, a_3, ..., a_N\}$, and so on until it stabilizes.

For the multinomial model, we sample (c, t, f) together, with the following posterior conditional probability, $P((c, t, f)|w_i, A_{-i}) \propto I(w_i = t, f)P(c_i = c|\theta^C, A_{-i})P(t|c, \theta^T, A_{-i})P(f|c, \theta^F, A_{-i})$. Take the second term of the above equation as an example. As a result of our choice of conjugate priors, the posterior distribution, $P(t|c, \theta^T, A_{-i})$, is also multinomial but with a different parameter $\alpha' \sim \text{Dirichlet}(\alpha^F + m_{c,f})$, where $m_{c,f}$ is the number of analyses that contain both inflectional class *c* and suffix *f*. Therefore, the second term can be reduced to a form with θ^T integrated out, $P(t_i = t|\mathbf{A}_{-i}, c) = \frac{\alpha^T + m_{c,f}}{\alpha_{+}^T + m_c}$. Similarly reductions can be done to other terms as well and putting together the results, we obtain the conditional probability for sampling (c, t, f) given the current word and morphological analyses of all other words, which is shown at the first row of Table 5.2.

For the lognormal model, we sample the inflection class *c* first, from the posterior distribution as discussed above. Then we sample (t, f) from its posterior distribution given the sampled *c* and morphological analyses of all other words. Again, given our choice of conjugate priors, the posterior distributions of stem *t* and suffix *f* are still lognormal with updated mean and variance. For the lognormal model, so as to update these parameters, we need to alternatively sample variance and mean from their own posterior distributions respectively. In practice, the updated mean and variance are sampled regarding the normal distribution of stem/suffix frequency's logarithms, $\mathfrak{L}(t/f)$. The specific sampling process is depicted in the second row of Table 5.2, for a comparison with the multinomial model³. We won't go into details of computing the updated parameters for sampling new mean and variance⁴, but it is worth noticing that the data samples are logarithms of frequencies.

The sampling of lognormal distributions is obviously much more complex and time-consuming than multinomial distributions. We are motivated to constrain the learning of generative models

³ Table 5.2 is constructed based on one of our reviewers' suggestion.

⁴ A detailed discussion can be seen in (Jordan, 2010)

with the acquisition outputs from the rule-based model, which runs very fast by itself. As we will show by experimental results in Section 6, the constrained learning is not only much faster but also significantly improves the performance.

6 Experiments

In this section, we run experiments on unsupervised learning of morphology and compare the approaches we describe in Section 3, 4 and 5. Following (Goldwater et al., 2011), we will learn morphological segmentations for verbs in the WSJ corpus with the input of raw text only. Given the Penn Treebank guidelines, we consider words associated with tags of 'VB', 'VBP', 'VBZ', 'VBD', 'VBN' or 'VBG' as verbs. Using the gold part-of-speech annotations, we extracted 137,899 verbs from the whole WSJ corpus which belong to 7,728 distinct word types. With heuristics based on part-of-speech tags and spellings, we automatically segment each verb into a stem, which cannot be empty, and a suffix, which may be empty, and use these segmentations as gold standards for evaluation.

Given the gold analysis of each word, the accuracy of a morphology model can be evaluated in two ways. For a type-based evaluation, we compute the accuracy as the percentage of correctly analyzed word types out of all distinct word types that are ever seen in the corpus. For a token-based evaluation, we compute the accuracy as the percentage of correctly analyzed tokens out of all occurrences of words in the whole corpus. In most previous work on unsupervised learning of morphology, only the type-based evaluation is reported. However, we agree with Goldwater et al. (2006) that the token-based evaluation gives more weights to the results of frequent words, thus reflects better the performance of each approach as applied to real text data.

Different forms of input

In formal study of morphology, the acquisition input is usually taken as the list of distinct word types. For example, as shown in Section 3.1, the bootstrapping algorithm measures contextual diversity with type frequency only. On the other hand, natural text data typically use most types of words more than once. Furthermore, when a model is trained with the input of distinct types only, each word occurrence of the same type will always receive the same analysis by the model. However, if a model is trained with real text data, then with a generative model, a word may receive different analyses on different occurrences.



Figure 3: Experiment with different forms of input.

First, we replicate the experiments in (Goldwater et al., 2006). The morphology model is multinomial as discussed in Section 4. As shown in Figure 3, the multinomial model learns well with the input of distinct word types, but poorly with the token-based input.

Inflectional classes



Figure 4: Experiment with different numbers of inflectional classes.

In the above experiment, the number of inflectional classes is set as 6 following (Goldwater et al., 2006). However, this choice of the number of inflectional classes is rather arbitrary. Therefore, we experiment with different settings of this parameter. As shown in Figure 4, different choices of inflectional classes do not make significant differences in training results.

Constrained Learning

As we described in Section 5, the sampling of the lognormal model is much more complex and time-consuming than the sampling of the multinomial model. Motivated by the concern of training efficiency, we propose to constrain the learning of generative models with the acquisition outputs from the rule-based model that we described in Section 3.1. Since the rule-based model takes input of raw text only and so the generative models, the combination of these two processes results in a totally unsupervised learning process as well. More specifically, suppose that we have acquired a set of suffixes containing *-es* and *-s* only. With constrained learning, the only possible segmentations we need to consider for word *porches* are *porch+es*, *porche+s*, and *porches+""*, instead of all the 7 possible segmentations of this 7-character word. In practice, we constrain the learning of generative models with 20 suffixes acquired by the bootstrapping algorithm.



Figure 5: Experiment with constrained learning of multinomial models.

As shown in Figure 5, the constrained learning converges much faster than its unconstrained counterpart. Since the bootstrapping algorithm is very fast by itself, taking almost no time compared to the training of generative models, the total training time is saved a lot. Moreover, even though

this method is motivated by the concern of training efficiency, it also significantly improves the performance. As clearly shown in Figure 5, the constrained learning achieves notably higher performance by both the type-based and token-based evaluations.

The generative model with lognormal likelihood

As discussed in Section 5, by replacing the multinomial likelihood with lognormal likelihood, we can take advantage of the particular statistical property of morphological units with a Bayesian approach; moreover, we can capture the statistical property of word frequency without the need of an additional generating process. We experiment with the lognormal model over different forms of input, and evaluate it by different criteria. Based on the above experimental results, in this experiment, we set the number of inflectional class as 1, and apply the constrained learning with 20 suffixes acquired by the bootstrapping algorithm.



Figure 6: Experiment with generative models of multinomial or lognormal distributoins.

Overall, the proposed model performs significantly better than the multinomial model, especially when trained with unprocessed text data. As shown in Figure 6-a, when they are both given the input of distinct types, the results of the lognormal and multinomial models are not distinguishable by the type-based evaluation. However, in contrast to the multinomial model, the lognormal model is able to learn from unprocessed text data as well. As shown in Figure 6-b, by the token-based evaluation, the proposed lognormal model achieves its best performance with the input of text data, which is much more accurate than the best of the multinomial model. It is interesting to observe that the proposed lognormal model is more accurate with the token-based evaluation when trained with token-based learning, but more accurate with the type-based evaluation when trained with type-based learning. This result pattern suggests that the proposed model is able to adapt to real data distributions by itself, without the need of an additional generative process.

Compare all three models

We have shown the acquisition outputs of the bootstrapping algorithm in Table 1, upon which we can build a rule-based segmentation model. In contrast to the learning progress of generative models, which will converge to a relatively steady state, we stop the acquisition process after 20 bootstrapping iterations following previous experiments. Furthermore, so as to compare the generative models with the rule-based model, for each generative model, we compute the average accuracy of its last 5 training iterations.

	input form	type-based evaluation	token-based evaluatoin
bootstrapping	either	83.59 %	64.04%
multinomial	type-based	79.98%	81.06%
lognormal	type-based	78.85%	73.10%
multinomial	token-based	42.36%	58.06%
lognormal	token-based	75.46%	87.79 %

Table 3: Compare all three models with different forms of input.

As shown in Table 3, the rule-based model achieves a type-based accuracy as high as 83.59%, significantly higher than any other generative model. However, by the token-based evaluation, the rule-based model performs rather bad. The highest token-based accuracy, 87.79%, is achieved by the lognormal generative model. No matter what form of input is fed to the multinomial model, this level of token-based accuracy cannot be achieved.

7 Conclusion and future work

In previous work on unsupervised learning of morphology, the long-tail pattern observed for the rankfrequency distribution of words, as well as of morphological units, is usually considered as following Zipf's law (power-law). We argue that the signature straight lines on logistic scales may suggest either power-law or lognormal. We have also discussed that both based on the idea of preferential attachment, the generative processes for generating Zipf's law and lognormal distributions have only subtle differences. The advantage of considering the long-tail distributions of morphological units as lognormal is so that we can utilize the statistical property in a Bayesian model. Moreover, given the multiplicative property of lognormal distributions, we can directly capture the long-tail distribution of word frequency without the need of an adaptor.

The experimental results show that the proposed model performs significantly better than other models in discussion, especially when it is evaluated by a token-based criterion that respects more of the real distribution of text data. Moreover, the proposed model can not only learn from the list of distinct word types, which can be handled by other models as well, but also from the unprocessed text data, which cannot be handled by other models. Especially, the proposed generative model is more accurate with the token-based evaluation when trained by token-based learning, and more accurate with the type-based evaluation when trained by type-based learning. This result pattern suggests that the proposed model is able to adapt to real data distribution by itself.

In this work, our primary goal is to provide an alternative perspective on modeling the long-tail distributions for morphology learning, rather than to develop a state-of-the-art morphology learning system. We are aware of recent work on morphology learning that utilize more extra information and achieve good results on more data. Extra information that has been shown to be useful for morphology model includes syntactic context (Lee et al., 2011), document boundaries (Moon et al., 2009) and so on. The proposed model has a potential to be developed as a more complex learning system, thus, in future work, we plan to extend our model to integrate these extra information and compare with more benchmark systems.

References

Chan, E. (2008). *Structures and distributions in morphology learning*. PhD thesis, University of Pennsylvania.

Downey, A. B. (2001). The structural cause of file size distributions. In *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, MASCOTS '01, Washington, DC, USA.

Goldwater, S., Griffiths, T., and Johnson, M. (2006). Interpolating between types and tokens by estimating power-law generators. In *NIPS*.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2011). Producing power-law distributions and damping word frequencies with two-stage language models. *Journal of Machine Learning Research*, 12(Jul):2335–2382.

Jordan, M. I. (2010). The conjugate prior for the normal distribution. Lecture notes on Stat260: Bayesian Modeling and Inference.

Lee, Y. K., Haghighi, A., and Barzilay, R. (2011). Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, Portland, Oregon, USA. Association for Computational Linguistics.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Miller, G. A. (1957). Some effects of intermittent silence. *American Journal of Psychology*, 70:311–314.

Mitzenmacher, M. (2004). A brief history of generative models for power law and lognormal distributions. *INTERNET MATHEMATICS*, 1:226–251.

Moon, T., Erk, K., and Baldridge, J. (2009). Unsupervised morphological segmentation and clustering with document boundaries. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 668–677, Stroudsburg, PA, USA. Association for Computational Linguistics.

Newman, M. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351.

Pareto, V. (1896). Cours d' Economie Politique. Droz, Geneva.

Perline, R. (1996). Zipf's law, the central limit theorem, and the random division of the unit interval. *Physical Review*, 54(1):220–223.

Pitman, J. and Yor, M. (1997). The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900.

Simon, H. A. (1955). On a class of skew distribution functions. Biometrika, 42(3/4):425-440.

Zhao, Q. and Marcus, M. (2011). Functional elements and pos categories. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1198–1206, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, Reading, MA.