Sentence Length

Gábor Borbély **András Kornai**
{borbely,kornai}@math.bme.hu
Department of Algebra, Budapest University of Technology and Economics

Abstract

The distribution of sentence length in ordinary language is not well captured by the existing models. Here we survey previous models of sentence length and present our random walk model that offers both a better fit with the data and a better understanding of the distribution. We develop a generalization of KL divergence, discuss measuring the noise inherent in a corpus, and present a hyperparameter-free Bayesian model comparison method that has strong conceptual ties to Minimal Description Length modeling. The models we obtain require only a few dozen bits, orders of magnitude less than the naive nonparametric MDL models would.

1 Introduction

Traditionally, statistical properties of sentence length distribution were investigated with the goal of settling disputed authorship (Mendenhall, 1887; Yule, 1939). Simple models, such as a "monkeys and typewriters" Bernoulli process (Miller, 1957) do not fit the data well, and this problem is inherited from n-gram Markov to n-gram Hidden Markov models, such as found in standard language modeling tools like SRILM (Stolcke et al., 2011). Today, length modeling is used more often as a downstream task to probe the properties of sentence vectors (Adi et al., 2017; Conneau et al., 2018), but the problem is highly relevant in other settings as well, in particular for the current generation of LSTM/GRU-based language models that generally use an ad hoc cutoff mechanism to regulate sentence length. The first modern study, interested in the entire shape of the sentence-length distribution, is Sichel (1974), who briefly summarizes the earlier proposals, in particular negative binomial (Yule, 1944), and lognormal (Williams, 1944), being rather critical of the latter:

> The lognormal model suggested by Williams and used by Wake must be rejected on several grounds: In the first place the number of words in a sentence constitutes a discrete variable whereas the lognormal distribution is continuous. Wake (1957) has pointed out that most observed log-sentence-length distributions display upper tails which tend towards zero much faster than the corresponding normal distribution. This is also evident in most of the cumulative percentage frequency distributions of sentence-lengths plotted on log-probability paper by Williams (1970). The sweep of the curves drawn through the plotted observations is concave upwards which means that we deal with sub-lognormal populations. In other words, most of the observed sentence-length distributions, after logarithmic transformation, are negatively skew. Finally, a mathematical distribution model which cannot fit real data –as shown up by the conventional χ^2 test– cannot claim serious attention. (Sichel, 1974, p. 26)

Sichel's own model is a mixture of Poisson distributions given as

$$\phi(r) = \frac{\sqrt{1-\theta}^{\gamma}}{K_\gamma(\alpha\sqrt{1-\theta})} \frac{(\alpha\theta/2)^r}{r!} K_{r+\gamma}(\alpha) \quad (1)$$

where K_γ is the modified Bessel function of the second kind of order γ. As Sichel notes, "a number of known discrete distribution functions such as the Poisson, negative binomial, geometric, Fisher's logarithmic series in its original and modified forms, Yule, Good, Waring and Riemann distributions are special or limiting forms of (1)".

While Sichel's own proposal certainly cannot be faulted on the grounds enumerated above, it still leaves something to be desired, in that the parameters α, γ, θ are not at all transparent, and the model lacks a clear genesis. In Section 2 of this article we present our own model aimed at remedying these defects and in Section 3 we analyze its properties. Our results are presented is Section 4. The relation between the sentence length model and grammatical theory is discussed in the concluding Section 5.

2 The random walk model

In the following Section we introduce our model of random walk(s). The predicted sentence length is basically the return time of these stochastic processes, i.e. the probability of a given length is the probability of the appropriate return time.

Let X_k be a random walk on $\mathbb{Z}$ and $X_k(t)$ the position of the walk at time t. Let $X_k(0) = k$ be the initial condition. The walk is given by the following parameters:

$$X_k(t+1) - X_k(t) = \begin{cases} -1 & \text{with probability } p_{-1} \\ 0 & \text{with probability } p_0 \\ 1 & \text{with probability } p_1 \\ 2 & \text{with probability } p_2 \end{cases} \tag{2}$$

The random walk is the sum of these independent steps. (2) is a simple model of valency (dependency) tracking: at any given point we may introduce, with probability p_2, some word with two open valences (e.g. a transitive verb), with probability p_1 one that brings one new valence (e.g. an intransitive verb or an adjective), with probability p_0 one that doesn't alter the count of open valencies (e.g. an adverbial), and with probability p_{-1} one that fills an open valency, e.g. a proper noun. For ease of presentation here we cut off at 2, making no provisions for ditransitives and higher arity verbs, but in actual numerical work (Section 3) we will relax this assumption. We also cut off at -1, making no provision for those cases where a single word can fill more than one valency, as in Latin *accusativus cum infinitivo* or (arguably) English equi. We discuss these cutoffs further in Sections 3.1 and 5. The return time is defined as

$$\tau_k = \min_{t \geq 0}\{t : X_k(t) = 0\} \tag{3}$$

In particular, τ_1 is the time needed to go from 1 → 0. We will calculate the probability-generating function to find the probabilities.

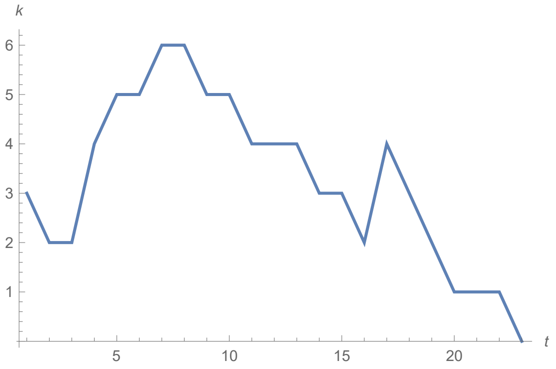

Figure 1: Sentence length is modeled as the return time of a random walk.

$$f(x) := \mathbb{E}\left(x^{\tau_1}\right) \tag{4}$$

The generating function of τ_k easily follows from τ_1, since τ_k is the sum of k independent copies of τ_1, so the generating function of τ_k is simply $f(x)^k$.

In order to calculate $f(x)$, we condition on the first step:

$$\begin{aligned} f(x) = p_{-1} \cdot x + & \quad \text{finishing in one step} \\ p_0 \cdot x \cdot f(x) + & \quad \text{wait } \tau_1 \text{ again} \\ p_1 \cdot x \cdot f(x)^2 + & \quad \text{wait } \tau_1 \text{ two times} \\ p_2 \cdot x \cdot f(x)^3 & \quad \text{wait } \tau_1 \text{ three times} \end{aligned} \tag{5}$$

Therefore, $f(x)$ is the solution of the following equation (solved for f, x is a parameter):

$$p_{-1} \cdot x + (p_0 \cdot x - 1) \cdot f + p_1 \cdot x \cdot f^2 + p_2 \cdot x \cdot f^3 = 0 \tag{6}$$

This can be solved with Cardano's formula. The probabilities are given by

$$\mathbb{P}(\tau_k = i) = [x^i] f(x)^k \triangleq \frac{1}{i!} \frac{\partial^i}{\partial x^i} f(x)^k \bigg|_{x=0} \tag{7}$$

(Here and in what follows, $[x^i]$ refers to the coefficient of x^n in the expansion of the function to the right of it.) For given parameters p_{-1}, p_0, p_1, p_2 and k, and a given i, one can evaluate these probabilities numerically, but we need a bit more analytical form. Let us define the following.

$$F(u) = p_{-1} + p_0 \cdot u + p_1 \cdot u^2 + p_2 u^3 \tag{8}$$

$$g(f) = \frac{f}{F(f)} \tag{9}$$

With these functions Equation 6 becomes $x = g(f(x))$, meaning that we are looking for the inverse function of g. One can see that $g(0) = 0$ and $g'(0) = 1/p_{-1} \neq 0$, therefore we can apply the Lagrange inversion theorem. Calculations detailed in the Appendix yield the following formula.

$$\mathbb{P}(\tau_k = i) = \frac{k}{i}[u^{i-k}]F^i(u) \qquad (10)$$

Since F is a polynomial, one can calculate its powers by polynomial multiplication and get $\mathbb{P}(\tau_k = i)$ by looking up the appropriate coefficient. Here k is an integer (discrete) model parameter and p_{-1}, p_0, p_1, p_2 are real (continuous) numbers. This makes the above mentioned probabilities *differentiable* in the continuous parameters.

We call the parameter k, the starting point of the random walk, the total valency. Note that $\tau_k \geq k$ with probability 1, therefore one cannot model the sentences shorter then k. To overcome this obstacle, we introduce the mixture model that consists of several models with various k values and coefficients for convex linear combination.

$$\mathbb{P}_{k_1,\alpha_1,k_2,\alpha_2,\ldots k_m,\alpha_m}(\tau = i) = \sum_{j=1}^{m} \alpha_j \cdot \mathbb{P}(\tau_{k_j} = i) \qquad (11)$$

where the parameters α_j are mixture coefficients; positive and sum up to 1, see Figure 2. Also every term in the mixture have different p_{-1}, p_0, p_1 and p_2 values (all positive and sum up to one). In this way, we can model the sentences with length at least $\min_j k_j$.

k_1	α_1	p^1_{-1}	p^1_0	p^1_1	p^1_2
k_2	α_2	p^2_{-1}	p^2_0	p^2_1	p^2_2
$\vdots$	$\vdots$		$\vdots$		
k_m	α_m	p^m_{-1}	p^m_0	p^m_1	p^m_2

Figure 2: Model parameters. The framed parameters are real, positive numbers and should sum up to 1.

It is easy to generalize our model to allow higher upward steps, i.e. p_3 for ditransitives or even higher steps for higher arity relations. The only technical constraint is that p_{-1} and p_0 should be positive, and no lower steps are allowed (no p_{-2}). This is also a reasonable assumption if a word can only fulfill one role in a sentence, a matter we return to in Section 5. Altogether, the number of upward steps is called *order* and it is an other hyper-parameter in our model.

Theoretically, there is no obstacle to have different number of p values to different k values. The model can be a heterogeneous mixture of random walks, where the individual processes can have different upward steps. But we did not investigate that possibility.

3 Model analysis

Here we introduce and analyze the experimental setup that we will use in Section 4 to fit our model to various datasets. The raw data is a set of positive integers, the sentence lengths, and their corresponding weights (absolute frequencies) $\{n_x\}_{x \in X}$. We call $n := \sum_{x \in X} n_x$ the *size* and X the *support* of the data. Since the model is differentiable in the continuous parameters (including the mixing coefficients), the direct approach would be to perform gradient descent on the dissimilarity as an objective function to find the parameters. With fixed valency parameters k_j this is a constrained optimization task $\text{dist}(\mathbb{P}_{\text{sample}}, \mathbb{P}_{\text{modeled}}) \to \min$.

In some cases, especially for smaller datasets, we might find it expedient to bin the data, for example (Adi et al., 2017) use bins (5-8), (9-12), (13-16), (17-20), (21-25), (26-29), (30-33), and (34-70). On empirical data (for English we will use the BNC[1] and the UMBC Webbase[2] and for other languages the SZTAKI corpus[3]) this particular binning leaves a lot to be desired. We discuss this matter in subsection 3.1, together with the choice of dissimilarity (figure of merit). An important consideration is that a high number of mixture components fit the data better but have more model parameters – this is discussed in subsection 3.2.

3.1 Length extremes

Short utterances are common both in spoken corpora and in written materials, especially in dialog intended to sound natural (see 2nd and 5th columns of Table 1). As is well known, people don't speak in complete sentences, and a great deal of the short material is the result of sluicing, zero anaphora, and similar cross-sentence ellipsis

[1] http://www.natcorp.ox.ac.uk

[2] https://ebiquity.umbc.edu/resource/html/id/351

[3] http://hlt.sztaki.hu/resources/webcorpora.html

phenomena (Merchant, 2001), with complete sentences such as imperatives like *Help!* comprising only a small portion of the data. In nonfiction, short strings are encountered overwhelmingly in titles, subtitles, and itemized lists, material that is hard to separate from actual sentences. Here we go around the problem by permitting in the mixture components with low total valency (small k at the start of the random walk).

dataset	< 5	>100	dataset	< 5	>100
BNC-A	7.2%	0.1%	Dutch	17.4%	1.1%
BNC-B	9.6%	0.1%	Finnish	14.1%	0.7%
BNC-C	8.8%	0.1%	Indonesian	11.3%	2.0%
BNC-D	25.9%	1.4%	Lithuanian	25.2%	1.1%
BNC-E	8.7%	0.1%	Bokmål	14.4%	1.1%
BNC-F	12.1%	0.2%	Nynorsk	8.7%	0.4%
BNC-G	11.2%	0.1%	Polish	23.3%	1.9%
BNC-H	14.5%	0.2%	Portuguese	22.7%	2.5%
BNC-J	15.2%	0.5%	Romanian	8.2%	3.1%
BNC-K	29.9%	0.2%	Serbian.sh	15.3%	1.9%
UMBC	3.7%	0.2%	Serbian.sr	33.7%	9.0%
Catalan	15.7%	2.8%	Slovak	12.4%	1.9%
Croatian	16.7%	2.1%	Spanish	14.7%	3.2%
Czech	13.7%	1.3%	Swedish	24.6%	0.8%
Danish	20.8%	1.1%			

Table 1: Distribution of short and long sentences

Especially on the long end (see columns 3 and 6 of Table 1) data becomes so sparse that some kind of binning is called for. Since the eight bins used by (Adi et al., 2017) actually ignore the very low (1-4) and very high (71+) ranges of the data, we will use ordinary deciles, setting the ten bins as the data dictates. In this regard, it is worth noting that in the 18 non-English corpora used in this study the low bin neglected by (Adi et al., 2017) contains on the average 17.4% of the data (variance 6.3%, low 8.1% on Romanian, high 33.7% on Serbian_sr) whereas on the high end the problem is much less severe: for example in UMBC 1.0%, and in the BNC only 0.8% would be ignored.

To cover 99.9% we need to consider only sentences up to a few hundred words (see column 5 of Table 2), and in the current study we applied a cutoff of 1,000 to be above 99.9% coverage in all cases while keeping compute time manageable. The last column of Table 2 shows the length of the longest sentence in each of the subcorpora considered. The original binning (cutoff at 71) would have resulted in coverage 95.7% on the average (variance 3.1%, low 84.9% Serbian_sr, high 98.8% for Nynorsk).

	dataset	number of	tolerance	mean	99.9%	max
		sentences	(in nats)	sentence length		
BNC-2.0 (English)	BNC-A	753442	9.847e-4	20.967	97	555
	BNC-B	362003	7.741e-3	20.650	96	365
	BNC-C	955486	9.494e-3	20.524	102	491
	BNC-D	6138	8.510e-2	16.366	228	466
	BNC-E	337370	5.000e-3	22.219	106	763
	BNC-F	527758	2.630e-2	19.351	130	2208
	BNC-G	478860	9.199e-3	18.753	106	435
	BNC-H	1185549	3.385e-2	18.841	118	950
	BNC-J	359352	7.940e-2	18.666	156	1100
	BNC-K	1086242	2.134e-1	12.784	116	918
	UMBC	136630947	2.442e-3	24.434	116	3052
SZTAKI corpus	Catalan	23927377	1.751e-3	27.496	384	5279
	Croatian	62196524	5.616e-3	23.975	369	8598
	Czech	30382696	5.147e-3	20.139	285	6081
	Danish	26687240	7.557e-3	18.593	296	16425
	Dutch	103958658	2.408e-3	19.135	296	16128
	Finnish	58104101	1.946e-3	15.538	237	5552
	Indonesian	13095607	1.231e-2	23.675	343	22762
	Lithuanian	81826291	1.184e-3	17.170	294	21857
	Bokmål	84375397	3.564e-3	19.199	281	14032
	Nynorsk	1393312	3.946e-3	18.836	175	1591
	Polish	72983880	8.508e-3	19.549	396	24353
	Portuguese	37953728	4.973e-2	25.365	448	9614
	Romanian	36211510	2.338e-2	29.466	473	54434
	Serbian.sh	35606837	4.531e-3	23.744	332	6800
	Serbian.sr	2023815	7.189e-3	37.736	862	6800
	Slovak	39633566	2.572e-3	21.759	402	24571
	Spanish	47673229	8.365e-4	29.305	471	29183
	Swedish	54218846	2.526e-3	16.468	315	8127

Table 2: Sentence length datasets. For tolerance see subsection 3.2

The prevailing tokenization convention, where punctuation is counted as equivalent to a full word, has an effect on the distribution, more perceptible at the low end. Besides this (and more subtle issues of tokenization, such as the treatment of hyphenation or of multiple punctuation) perhaps the most important factor influencing sentence length is morphological complexity, since in highly agglutinating languages a single word is sufficient for what would require a multiword sentence in English, as in Hungarian *elvihetlek* 'I can give you a ride'.

Since the number of datapoints is high, ranging from 1.3M (Nynorsk) to 136.6M (UMBC), the conventional χ^2 test does not provide a good figure of merit on the original data (no fit is ever significant, especially as there is a lot of variation at the high end where only few lengths are extant), nor on the binned data, where every fit is highly significant.

A better choice is the Kullback–Leibler divergence, but this still suffers from problems when the supports of the distributions do not coincide. In our case we have this problem both at the low end, where the model predicts $\mathbb{P}(\tau = i) = 0$ for $i < k$, and at the high end, where we predict positive (albeit astronomically small) probabilities of arbitrarily long sentences. To remedy this defect, we define generalized KL divergence, gKL, as follows.

Definition 3.1 (Motivated by Theorem A.2.)**.** *Let $\mathbb{P}$ and $\mathbb{Q}$ be probability measures over the same measurable space (X, Σ) that are both absolutely continuous with respect to a third measure $\mathrm{d}x$, and let λ be $\mathbb{P}(\operatorname{supp}(\mathbb{P}) \cap \operatorname{supp}(\mathbb{Q}))$. Then*

$$gKL(\mathbb{P}, \mathbb{Q}) := -\lambda \cdot \ln \lambda + \int\limits_{\operatorname{supp}(\mathbb{P}) \cap \operatorname{supp}(\mathbb{Q})} \mathbb{P}(x) \cdot \ln \frac{\mathbb{P}(x)}{\mathbb{Q}(x)} \, \mathrm{d}x \tag{12}$$

Clearly, gKL reduces to the usual KL divergence if the support of the distributions coincide. The high end of the distribution could be ignored, at least for English, at the price of losing less than 0.1% of the data, but ignoring the short sentences, 14.4% of the BNC, is hard to countenance. As a practical matter this means we needed to bring in mixture components with total valency $k < 4$, and these each bring 4 parameters (the mixture weight and 3 p_i values) in tow. Obviously, the more components we use, the better the fit will be, so we need to control the trade-off between these. In subsection 3.2 we introduce a method derived from Bayesian model comparison (MacKay, 2003) that will remedy the zero modeled probabilities and answer the model complexity trade-off.

3.2 Bayesian model comparison

If a dataset D has support X, with $n_x > 0$ being the number that length x occurred, the data size is $|D| = \sum_{x \in X} n_x$ and the observed probabilities are $p_x := \frac{n_x}{|D|}$. Let $\mathcal{H}_i \subseteq \mathbb{R}^d$ be i^{th} model in some list of models. Each model is represented by a parameter vector $\mathbf{w}_i \in \mathcal{H}_i$ in the parameter space, and $\operatorname{supp} \mathcal{H}_i = \{x \mid \mathbb{P}(x \mid \mathcal{H}_i) > 0\}$ is not necessarily equal to X. Clearly, different $\mathcal{H}_i$ may have different support, but a given model has the same support for every $\mathbf{w}_i$. Model predictions are given by $\mathbb{Q}_{\mathbf{w}_i}(x) := \mathbb{P}(x \mid \mathbf{w}_i, \mathcal{H}_i)$, and the **evidence** the i^{th} model has is

$$\mathbb{P}(\mathcal{H}_i \mid D) = \frac{\mathbb{P}(D \mid \mathcal{H}_i) \cdot \mathbb{P}(\mathcal{H}_i)}{\mathbb{P}(D)} \tag{13}$$

If one supposes that no model is preferred over any other models ($\mathbb{P}(\mathcal{H}_i)$ is constant) then the decision simplifies to finding the model that maximizes

$$\mathbb{P}(D \mid \mathcal{H}_i) = \int_{\mathcal{H}_i} \mathbb{P}(D \mid \mathbf{w}_i, \mathcal{H}_i) \cdot \mathbb{P}(\mathbf{w}_i \mid \mathcal{H}_i) \, \mathrm{d}\mathbf{w}_i \tag{14}$$

We make sure that no model parameter is preferred by setting a uniform prior:

$$\mathbb{P}(\mathbf{w}_i \mid \mathcal{H}_i) = 1 / \left(\int_{\mathcal{H}_i} 1 \, \mathrm{d}\mathbf{w}_i \right) = 1/\operatorname{Vol}(\mathcal{H}_i) \tag{15}$$

We estimated this integral with Laplace's method by introducing $f(\mathbf{w}_i) := -\frac{1}{|D|} \ln \mathbb{Q}_{\mathbf{w}_i}(D)$, i.e. the cross entropy (measured in nats).

$$\begin{aligned} \mathbb{P}(D \mid \mathbf{w}_i, \mathcal{H}_i) &= \prod_{x \in X} \mathbb{Q}_{\mathbf{w}_i}(x)^{n_x} \\ f(\mathbf{w}_i) &= -\sum_{x \in X} p_x \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x) \end{aligned} \tag{16}$$

Taking $-\frac{1}{|D|} \ln(\bullet)$ of the evidence amounts to minimizing in i the following quantity:

$$\begin{gathered} f(\mathbf{w}_i^*) + \frac{1}{|D|} \cdot \ln \operatorname{Vol}(\mathcal{H}_i) + \\ \frac{1}{2|D|} \ln \det f''(\mathbf{w}_i^*) + \frac{d}{2|D|} \cdot \ln \frac{|D|}{2\pi} \end{gathered} \tag{17}$$

where d is the dimension of $\mathcal{H}_i$ (number of parameters), f'' is the Hessian and $\mathbf{w}_i^* = \arg\min_{\mathbf{w}_i \in \mathcal{H}_i} f(\mathbf{w}_i)$ for a given i. Since the theoretical optimum of $f(\mathbf{w}_i)$ is the entropy of the data ($\ln 2 \cdot \mathrm{H}(D)$), we subtract this quantity from Equation 17 so that the term $f(\mathbf{w}_i)$ becomes the relative entropy (measured in nats) with a theoretical minimum of 0.

We introduce an augmented model to deal with the datapoints where $\mathbb{Q}_{\mathbf{w}_i}(x) = 0$.

$$\overline{\mathbb{Q}}_{\mathbf{w}_i, \mathbf{q}}(x) := \begin{cases} \lambda \mathbb{Q}_{\mathbf{w}_i}(x) & \text{if } \mathbb{Q}_{\mathbf{w}_i}(x) > 0 \\ (1 - \lambda) q_x & \text{if } n_x > 0, \mathbb{Q}_{\mathbf{w}_i}(x) = 0 \end{cases} \tag{18}$$

where

$$\lambda = \sum_{x \in X \cap \operatorname{supp}(\mathcal{H}_i)} p_x \qquad \text{covered probability}$$

$$1 - \lambda = \sum_{x \in X \setminus \operatorname{supp}(\mathcal{H}_i)} p_x \qquad \text{uncovered probability}$$

The newly introduced model parameters $\mathbf{q} = (q_x)_{x \in X \setminus \operatorname{supp}(\mathcal{H}_i)}$ are also constrained: they have to be positive and sum up to one, i.e. inside the probability simplex. After finding the optimum of

$\mathbf{q}$ and modifying Equation 17 with the auxiliary terms and subtracting the entropy of the data ($\ln 2 \cdot \mathrm{H}(D)$) as discussed above, one gets:

$$\begin{aligned}
&-\lambda \cdot \ln \lambda + \sum_{x \in X \cap \operatorname{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)} + \\
&\frac{1}{|D|} \cdot (\ln \mathrm{Vol}(\mathcal{H}_i) + \ln \mathrm{Vol}(\text{aux. model})) + \\
&\frac{1}{2|D|} \cdot \ln \det\left(\text{model Hessian}\right) + \\
&\frac{1}{2|D|} \cdot \ln \det\left(\text{aux. model Hessian}\right) + \\
&\frac{d'}{2|D|} \cdot \ln \frac{|D|}{2\pi} \qquad (19)
\end{aligned}$$

where d' is the original model dimension plus the auxiliary model dimension. One possible use (or abuse) of auxilary parameters would be to directly (nonparametrically) model the low end of the length distribution. But, as we shall see in Section 4, the parametric models actually do better. To see what is going in, let us consider the asymptotic behavior of models.

For sufficiently large corpora ($|D| \to \infty$) all but the first term will be negligible, meaning that the most precise model (in terms of gKL divergence) wins regardless of model size. One way out would be to choose an 'optimum corpus size' (Zipf, 1949), a move that has already drawn strong criticism in Powers (1998) and one that would amount to little more than the addition of an extra hyperparameter to be set heuristically.

Another, more data-driven approach is based on the observation that corpora have *inherent noise*, measurable as the KL divergence between a random subcorpus and its complement (Kornai et al., 2013) both about the same size (half the original). Here we need to take into account the fact that large sentence lengths appear with frequency 1 or 0, so subcorpora D_1 and $D_2 = D \setminus D_1$ will not have the exact same support as the original, and we need to use symmetrized gKL: the **inherent noise** δ_D of a corpus D is $\frac{1}{2}(gKL(D_1, D_2) + gKL(D_2, D_1))$, where D_1 and D_2 are equal size subsets of the original corpus D, and the gKL divergence is measured on their empirical distributions.

δ_D is largely independent of the choice of subsets D_1, D_2 of the original corpus, and can be easily estimated by randomly sampled D_is. To the extent crawl data and classical corpora are sequentially structured (Curran and Osborne, 2002), we sometimes obtain different noise estimates based on random D_i than from comparing the first to the second half of a corpus, the procedure we followed here. In the Minimum Description Length (MDL) setting where this notion was originally developed it is obvious that we need not approximate corpora to a precision better than δ, but in the Bayesian setup that we use here matters are a bit more complicated.

Definition 3.2. *For $\delta > 0$ let*

$$gKL_\delta(\mathbb{P}, \mathbb{Q}) := \max(0, gKL(\mathbb{P}, \mathbb{Q}) - \delta) \quad (20)$$

*For a sample $\mathbb{P}$ with inherent noise δ, a model $\mathbb{Q}$ is called **tolerable** if $gKL_\delta(\mathbb{P}, \mathbb{Q}) = 0$*

If gKL_δ is used instead of gKL in Equation 19 then model size d becomes important. If a model fits within δ then the first term becomes zero and for large $|D|$ values the number of model parameters (including auxiliary parameters) will dominate the evidence. The limiting behavior of our evidence formula, with tolerance for inherent noise, is determined by the following observations:

1. Any tolerable model beats any non-tolerable one.

2. If two models are both tolerable and have different number of model parameters (including auxiliary model), then the one with the fewer parameters wins.

3. If two models are both tolerable and have the same number of parameters, then the model volume and Hessian decides.

An interesting case is when no model can reach the inherent noise – in this case we recover the original situation where the best fit wins, no matter the model size.

4 Results

A single model $\mathcal{H}_i$ fit to some dataset is identified by its *order*, defined as the number of upward steps the random walk can take at once: $1, 2$ or 3, marked by the number before the first decimal; and its *mixture*, a non-empty subset of $\{1, 2, 3, 4, 5\}$ that can appear as k: valency of a single component. For example **1.k1.2.4** marks order 1 and k mixture: $\{1, 2, 4\} \subseteq \{1, 2, 3, 4, 5\}$. Altogether, we trained $3 \times 31 = 93$ locally optimal models for each dataset and compared them with Equation 19, except that gKL_δ is used with the appropriate tolerance.

We computed $\mathbf{w}_i^*$ with a (non-batched) gradient descent algorithm.[4] We used Adagrad with initial learning rate $\eta = 0.9$, starting from uniform p and α values, and iterated until every coordinate of the gradient fell within $\pm 10^{-3}$. The gradient descent typically took $10^2 - 10^3$ iterations to reach a plateau, but about .1% of the models were more sensitive and required a smaller learning rate $\eta = 0.1$ with more (10k) iterations.

4.1 Validation

The model comparison methodology was first tested on artificially generated data. We generated 1M+1M samples of pseudo-random walks with parameters: $p_{-1} = 0.5, p_0 = p_1 = 0.25$ (at most one step upward) and $k = 3$ (no mixture) and obtained the inherent noise and length distribution. The inherent noise was about 3.442e-4 nats. We trained all 93 models and compared them as described above.

The validation data size is $2 \cdot 10^6$ but we also replaced $|D|$ with a hyper-parameter n in Equation 19. This means that we faked the sample to be bigger (or smaller) with the same empirical distribution. We did this with the goal of imitating the 'optimum corpus size' as an adverse effect.

As seen on Table 3 the true model wins. We also tested the case when the true model was simply excluded from the competing models. In this case, the tolerance is needed to ensure a stable result as $n \to \infty$.

1.k3 artificial data	best parameters for various n values					
	1k	10k	100k	1M	10M	1G
with tolerance	3.k1-5	1.k3	1.k3	1.k3	1.k3	1.k3
w/o tolerance	3.k1-5	1.k3	1.k3	1.k3	1.k3	1.k3
w tolerance, -true	3.k1-5	2.k4	2.k4	2.k4	2.k4	2.k4
w/o tolerance, -true	3.k1-5	2.k4	1.k2.3	1.k2.3	1.k2.3	1.k3-5

Table 3: Optimal models for artificially generated data (1.k3) for various n values.

As there are strong conceptual similarities between MDL methods and the Bayesian approach (MacKay, 2003), we also compared the models with MDL, using the same locally optimal parameters as before, but encoding them in bits. To this end we used a technique from (Kornai et al., 2013) where all of the continuous model parameters are discretized on a log scale unless the discretization error exceeds the tolerance. The model with the least number of bits required wins if it fits within tolerance. (The constraints are hard-coded in this model, meaning that we re-normalized the parameters after the discretization.) In the artificial test example, the model 1.k3 wins, which is also the winner of the Bayesian comparison. If the true model is excluded, the winner is 1.k2.3. Further MDL results will be discussed in Section 4.4.

4.2 Empirical data

Let us now turn to the natural language corpora summarized in Table 2. Not only are the webcrawl datasets larger than the BNC sections, but they are somewhat noisier and have suspiciously long sentences. To ease the computation, we excluded sentences longer than $1,000$ tokens. This cutoff is always well above the 99.9th percentile given in the next to last column of Table 2. The results, summarized in Table 4, show several major tendencies.

First, most of the models (151 out of 174) fit sentence length of the entire subcorpus better than the empirical distribution of the first half would fit the distribution of the second half. When this criterion is *not* met for the best model, i.e. the gKL distance of the model from the data is above the internal noise, the ill-fitting model form is shown in *italics*.

Second, this phenomenon of not achieving tolerable fit is seen primarily (16 out of 29) in the first column of Table 4, corresponding to a radically undersampled condition $n = 1,000$, and (7 out of 29) to a somewhat undersampled condition $n = 10,000$.

Third, and perhaps most important, for sufficiently large n the Bayesian model comparison technique we advocate here actually selects rather simple models, with order 1 (no ditransitives, a matter we return to in Section 5) and only one or two mixture components. We emphasize that 'sufficiently large' is still in the realistic range, one does not have to take the limit $n \to \infty$ to obtain the correct model. The last two columns (gigadata and infinity) always coincide, and in 21 of the 29 corpora the 1M column already yield the same result.

Given that tolerance is generally small, less than 0.66 bits even in our noisiest corpus (BNC-K), we didn't expect much change if we perform

[4] You can find all of our code used for training and evaluating at `https://github.com/hlt-bme-hu/SentenceLength`

the model comparison without using Equation 20. Unsurprisingly, if we reward every tiny improvement in divergence, we get more models (159 out of 174) within the tolerable range – those outside the tolerance limit are again given in italics in Table 6. But we pay a heavy price in model complexity: the best models (in the last two columns) are now often second order, and we have to countenance a hyperparameter n which matters (e.g. for Polish).

dataset	best parameters for various n values					
	1k	10k	100k	1M	1G	∞
BNC-A	*3.k1-5*	*3.k2-5*	1.k4.5	1.k4.5	1.k4.5	1.k4.5
BNC-B	*3.k1-5*	*3.k1-5*	1.k1.5	1.k1.5	1.k1.5	1.k1.5
BNC-C	3.k2-5	3.k2-5	3.k2-5	1.k1.4	1.k1.4	1.k1.4
BNC-D	3.k2.3.5	3.k2.3.5	3.k2.3.5	1.k2	1.k2	1.k2
BNC-E	*3.k1.3-5*	*3.k1.3-5*	1.k2.5	1.k2.5	1.k2.5	1.k2.5
BNC-F	3.k3.4.5	3.k3.4.5	3.k3.4.5	1.k3	1.k3	1.k3
BNC-G	3.k1-5	3.k1-5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
BNC-H	*3.k2.4.5*	3.k3.4.5	1.k4	1.k4	1.k4	1.k4
BNC-J	3.k2.3.4	3.k2.3.4	3.k2.5	1.k2	1.k2	1.k2
BNC-K	3.k1-5	3.k1-5	1.k2	1.k2	1.k2	1.k2
UMBC	*3.k1.3-5*	*3.k1.3-5*	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Catalan	*3.k2-5*	*3.k2-5*	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Croatian	3.k3.4.5	3.k3.4.5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Czech	*3.k4.5*	3.k1.3.5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Danish	*3.k1-5*	3.k1.3.5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Dutch	*3.k1-5*	*3.k3.4.5*	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Finnish	*3.k1.3.5*	1.k2.4	1.k2.4	1.k2.4	1.k2.4	1.k2.4
Indonesian	3.k1-5	3.k1-5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Lithuanian	*3.k2.3.4*	*3.k2.3.4*	1.k2.3	1.k2.3	1.k2.3	1.k2.3
Bokmål	3.k2.4.5	3.k2.4.5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Nynorsk	*3.k1-5*	1.k2.5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Polish	3.k2-5	3.k2-5	3.k2-5	3.k2-5	1.k2.5	1.k2.5
Portuguese	3.k2.3.5	3.k2.3.5	1.k2	1.k2	1.k2	1.k2
Romanian	3.k1.3-5	3.k1.3-5	1.k5	1.k5	1.k5	1.k5
Serbian.sh	*3.k1.2.4.5*	3.k2.3.5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Serbian.sr	*3.k2-5*	3.k2.3.4	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Slovak	*3.k2.4.5*	3.k2-5	1.k2.5	1.k2.5	1.k2.5	1.k2.5
Spanish	*3.k2.4.5*	1.k2.3	1.k2.3	1.k2.3	1.k2.3	1.k2.3
Swedish	1.k2.4	1.k2.4	1.k2.4	1.k2.4	1.k2.4	1.k2.4

Table 4: Optimal models with tolerance for inner noise. *Ill-fitting models* are marked with italics.

4.3 Previous models

We also compared previous or baseline sentence length models with our new model. The hyper-parameters of the *bins* model are the bins themselves. The distribution over the bins are the continuous model-parameters. For m bins: $[1, b_1), [b_1, b_2), \ldots [b_{m-1}, \infty)$, the probability distribution $\mathbb{P}(b_i \leq X < b_{i+1}) = q_i$ is to be optimized. This model has $m - 1$ free parameters (model dimension) and its model volume is the volume of a probabilistic m-simplex. No auxiliary model is required.

We also trained[5] and compared Sichel's model (Equation 1) with our method. In this case α and θ are the model-parameters and γ was a non-trained hyper-parameter. In Sichel (1974) it was fixed $\gamma = -\frac{1}{2}$, we trained $\gamma \in \{-0.5, -0.4\}$, the higher γ value was usually better. Again no auxiliary model was needed.

dataset	Sichel	binned	randwalk	δ
BNC-A	*3.554e-2*	*1.489e-2*	4.409e-4	9.847e-4
BNC-B	*6.212e-2*	*1.274e-2*	7.215e-3	7.741e-3
BNC-C	*4.861e-2*	*1.431e-2*	6.989e-3	9.494e-3
BNC-D	*9.917e-2*	8.387e-2	5.945e-2	8.510e-2
BNC-E	*6.976e-2*	*2.251e-2*	4.353e-3	5.000e-3
BNC-F	*3.153e-2*	2.196e-2	2.270e-2	2.630e-2
BNC-G	*2.598e-2*	*1.495e-2*	5.762e-3	9.199e-3
BNC-H	*4.765e-2*	3.265e-2	3.106e-2	3.385e-2
BNC-J	3.048e-2	6.854e-2	2.946e-2	7.940e-2
BNC-K	6.583e-2	1.388e-1	3.899e-2	2.134e-1
UMBC	*6.584e-2*	*2.615e-2*	1.390e-3	2.442e-3
Catalan	*1.389e-1*	*6.102e-2*	9.382e-4	1.751e-3
Croatian	*1.131e-1*	*4.604e-2*	2.063e-3	5.616e-3
Czech	*5.857e-2*	*3.687e-2*	2.563e-3	5.147e-3
Danish	*1.618e-1*	*3.072e-2*	2.772e-3	7.557e-3
Dutch	*4.232e-1*	*3.447e-2*	1.391e-3	2.408e-3
Finnish	*9.968e-2*	*2.830e-2*	1.659e-3	1.946e-3
Indonesian	*2.159e-1*	*5.017e-2*	1.390e-3	1.231e-2
Lithuanian	-	*3.113e-2*	6.637e-4	1.184e-3
Bokmål	-	*3.332e-2*	3.515e-3	3.564e-3
Nynorsk	-	*2.830e-2*	3.757e-3	3.946e-3
Polish	-	*4.078e-2*	1.518e-3	8.508e-3
Portuguese	-	*5.133e-2*	4.514e-2	4.973e-2
Romanian	-	*6.539e-2*	1.579e-2	2.338e-2
Serbian.sh	-	*4.676e-2*	1.346e-3	4.531e-3
Serbian.sr	-	*1.389e-1*	6.971e-3	7.189e-3
Slovak	-	*4.344e-2*	2.184e-3	2.572e-3
Spanish	-	*6.501e-2*	7.718e-4	8.365e-4
Swedish	-	*2.652e-2*	2.310e-3	2.526e-3

Table 5: Best of the models and their fit. *Ill-fitting models* are marked with italics.

As can be seen, the fit is always improved (on the average by 40%) from the mixture Poisson to the binned model, and the random walk model further improves from the binned (on the average by 70%). More important, the mixture Poisson model never, the binned model rarely, but the random walk model always approximates the data better than its inner noise. Altogether the random walk models always outperforms the other two, but not always for the same reason. In the case of bins, the fit was poor and only the fine-grained bins per-

[5] Optimizing the mixture Poisson coefficients took orders of magnitude more time than optimizing the other models. The difficulties come from computing the derivatives of Bessel functions. At the time of going to press still about a third of the values are missing – by the time of the meeting these will be published at `https://github.com/hlt-bme-hu/SentenceLength`

formed within inherent noise. Note that none of our parametric models use mode than 11 parameters, which makes only systems with 12 or fewer bins competitive.

In case of Equation 1, Sichel already mentions that the fit is satisfactory only with binned probabilities, i.e. on a dumbed down distribution with 4-5 data points aggregated into one. This classic model has only 2 parameters, which would make it very competitive for large inherent noise or small data size, but neither is the case here.

4.4 MDL approach

Finally, let us consider the MDL results given in Table 7. These are often (9 out of 29 subcorpora) consistent with the results obtained using Equation 20, but never with those obtained without considering inherent noise to be a factor. Remarkably, we never needed more than 6 bits quantization, consistent with the general principles of Google's TPUs (Jouppi et al., 2017) and is in fact suggestive of an even sparser quantization regime than the eight bits employed there.

For a baseline, we discretized the naive (non-parametric) model in the same way. Not only does the quantization require on the average two bits more, but we also have to countenance a considerably larger number of parameters to specify the distribution within inherent noise, so that the random walk model offers a size savings of at least 95.3% (BNC-A) to 99.7% (Polish).

With the random walk model, the total number of bits required for characterizing the most complex distributions (66 for BNC-A and 60 for Spanish) appears to be more related to the high consistency (low internal noise) of these corpora than to the complexity of the length distributions.

5 Conclusion

At the outset of the paper we criticized the standard mixture Poisson length model of Equation 1 for lack of a clear genesis – there is no obvious candidate for 'arrivals' or for the mixture. In contrast, our random walk model is based on the suggestive idea of total valency 'number of things you want to say', and we see some rather clear methods for probing this further.

First, we have extensive lexical data on the valency of individual words, and know in advance that e.g. color adjectives will be dependent on nouns, while relational nouns such as *sister* can bring further nouns or NPs. Combining the lexical knowledge with word frequency statistics is somewhat complicated by the fact that a single word form may have different senses with different valency frames, but these cause no problems for a statistical model that convolves the two distributions.

dataset	best parameters for various n values				
	1k	10k	100k	1M	1G
BNC-A	*3.k1-5*	1.k4.5	1.k4.5	1.k1-5	1.k1-5
BNC-B	*3.k1-5*	1.k2.3.5	2.k4.5	2.k4.5	2.k4.5
BNC-C	3.k2-5	1.k2.4.5	1.k2.4.5	1.k2.4.5	1.k2.4.5
BNC-D	3.k3.4	1.k2.5	2.k2.5	2.k2.5	2.k2.5
BNC-E	*3.k1.3-5*	1.k4.5	1.k4.5	1.k4.5	1.k4.5
BNC-F	3.k3-5	1.k2.4.5	1.k2.4.5	1.k2.4.5	1.k2.4.5
BNC-G	3.k1-5	1.k4.5	1.k2.4.5	1.k2.4.5	2.k2.4.5
BNC-H	3.k3-5	1.k4.5	2.k2.4.5	2.k2.4.5	2.k2.4.5
BNC-J	3.k1-5	1.k2.4.5	1.k2.4.5	1.k2.4.5	1.k2.4.5
BNC-K	3.k2-5	3.k2-5	1.k2.4.5	1.k2.4.5	1.k2.4.5
UMBC	*3.k1.3-5*	1.k2.4	1.k2.4.5	1.k2.4.5	1.k2.4.5
Catalan	*3.k2-5*	*3.k2-5*	1.k2.4	1.k1.3-5	1.k1.3-5
Croatian	3.k3-5	1.k2.3	1.k2.3	1.k3-5	1.k3-5
Czech	3.k2-5	3.k3-5	1.k2.3	1.k1.3-5	1.k1.3-5
Danish	*3.k1-5*	1.k2.3	1.k1.2.4.5	1.k1.2.4.5	3.k2-5
Dutch	*3.k1-5*	1.k2.4	1.k3.4	1.k1-5	1.k1-5
Finnish	*3.k1.3.5*	1.k1.3.4	1.k1.3.4	1.k1.3-5	1.k1.3-5
Indonesian	3.k1-5	1.k3.5	1.k3-5	1.k3-5	1.k3-5
Lithuanian	*3.k2.3.4*	1.k2.3	1.k2-5	1.k2-5	1.k2-5
Bokmål	3.k2.4.5	3.k2.4.5	1.k1.3-5	1.k1.3-5	1.k1.3-5
Nynorsk	*3.k1-5*	1.k2.4.5	1.k1-5	1.k1-5	1.k1-5
Polish	3.k2-5	3.k2-5	1.k1.4.5	1.k2-5	1.k2-5
Portuguese	3.k2.4.5	1.k2.3	1.k3.4	1.k3.4	1.k3.4
Romanian	3.k2.4.5	1.k2.4	1.k2.3.4	1.k2.3.4	1.k2.3.4
Serbian.sh	*3.k1.2.4.5*	1.k2.4	1.k3.4	1.k2-5	1.k2-5
Serbian.sr	*3.k2-5*	1.k4.5	1.k4.5	1.k4.5	1.k4.5
Slovak	*3.k2.4.5*	1.k2.3	1.k1.3-5	1.k1.3-5	1.k1.3-5
Spanish	*3.k2.4.5*	1.k2.3	1.k1.3.5	1.k1.3.5	1.k1.3.5
Swedish	1.k2.3	1.k2.3	1.k1-5	1.k1-5	1.k1-5

Table 6: Optimal models without tolerance. *Ill-fitting models* are marked with italics.

Second, thanks to Universal Dependencies[6] we now have access to high quality dependency treebanks where the number of dependencies running between words $w_1, \ldots, w_k$ and $w_{k+1} \ldots w_n$, the y coordinate of our random walk at k, can be explicitly tracked. Using these treebanks, we could perform a far more detailed analysis of phrase or clause formation than we attempted here, e.g. by systematic comparison of the learned p_1 and p_2 values with the observable proportion of intransitive and transitive verbs and relational nouns. Ditransitives are rare (in fact they usually make up less than 2% of the verbs) and we think these can be eliminated entirely (Kornai, 2012) without loss

[6] `http://universaldependencies.org`

dataset	mq	nq	tb	opt	% size
BNC-A	6	7	66	1.k4.5	4.69
BNC-B	4	5	40	2.k2.5	4.65
BNC-C	3	5	36	1.k1.2.4	3.32
BNC-D	2	3	6	1.k2	1.29
BNC-E	4	5	32	1.k2.5	3.56
BNC-F	2	5	16	1.k2.5	1.13
BNC-G	3	5	24	1.k1.2	2.45
BNC-H	2	4	16	1.k2.5	1.36
BNC-J	2	4	6	1.k2	0.51
BNC-K	2	3	6	1.k2	0.63
UMBC	4	7	44	1.k4.5	0.88
Catalan	5	7	40	1.k2.5	0.57
Croatian	4	6	32	1.k2.4	0.53
Czech	3	6	24	1.k2.5	0.41
Danish	3	6	24	1.k2.4	0.41
Dutch	5	7	40	1.k2.5	0.57
Finnish	4	7	48	1.k1.2.3	0.69
Indonesian	4	5	32	1.k2.4	0.66
Lithuanian	4	7	32	1.k2.3	0.46
Bokmål	3	7	30	2.k2.5	0.43
Nynorsk	4	6	32	1.k2.3	1.14
Polish	2	5	16	1.k2.5	0.32
Portuguese	3	5	18	1.k4	0.36
Romanian	3	5	24	1.k1.2	0.48
Serbian.sh	4	6	32	1.k2.4	0.53
Serbian.sr	4	5	32	1.k2.5	0.64
Slovak	5	6	40	1.k2.3	0.67
Spanish	6	7	60	1.k3.4	0.86
Swedish	5	7	40	1.k2.3	0.57

Table 7: Optimal models with MDL comparison (with tolerance). mq: Model quantization bits. nq: naive/nonparametric quantization bits. tb: total bits. opt: optimal model configuration. %size: size of random walk model as percentage of size of nonparametric model.

of generality. The same kind of analysis could be attempted for other grammatical formalisms like type-logical grammars, which make tracking the open arguments an even more attractive proposition, but unfortunately these lack large parsed corpora. Another significant issue with formalisms other than UD is that the cross-linguistic breadth of parsed corpora is minute – do we want to base general conclusions of the type attempted here, linking predicate/argument structure to sentence length, on English alone?

Third, we can extend the analysis in a typologically sound manner to morphologically more complex languages. Using a morphologically analyzed Hungarian corpus (Oravecz et al., 2014) we measured the per-word morpheme distribution and per-sentence word distribution. We found that the random sum of 'number of words in a sentence' independent copies of 'number of morphemes in a word' estimates the per-sentence morpheme distribution within inherent noise. To the extent these results can be replicated for other morphologically complex languages (again UD morphologies[7] offer the best testbed, though a lot remains to be done for ensuring homogeneity) problems like six-word 'I can give you a ride' versus one-word *elvihetlek* disappear.

Another avenue of research alluded to above would be the study of subject- and object-control verbs and infinitival constructions, where single nouns or NPs can fill more than one open dependency. This would complicate the calculations in Equation 5 in a non-trivial way. We plan to extend our mathematical model in a future work, but it should be clear from the foregoing that sentences exhibiting these phenomena are so rare as to render unlikely any prospect of improving the statistical model by means of accounting for these. This is not to say that control phenomena are irrelevant to grammar – but they are likely 'within the noise' for statistical length modeling.

One of the authors (Kornai and Tuza, 1992) already suggested that the number of dependencies open at any given point in the sentence must be subject to limitations of short-term memory (Miller, 1956) – this may act as a reflective barrier that keeps asymptotic sentence length smaller than the pure random walk model would suggest. In particular, Bernoulli and other well-known models predict exponential decay at the high end, whereas our data shows polynomial decay proportional to n^{-C}, with C somewhere around 4 (in the $3 - 5$ range). This is one area where our corpora are too small to draw reliable conclusions, but overall we should emphasize that corpora already collected (and in the case of UD treebanks, already analyzed) offer a rich empirical field for studying sentence length phenomena, and the model presented here makes it possible to use statistics to shed light on the underlying grammatico-semantic structure.

Acknowledgments

The presentation greatly benefited from the remarks of the anonymous reviewers. Research partially supported by National Research, Development and Innovation Office (NKFIH) grants #120145: Deep Learning of Morphological Structure and NKFIH grant #115288: Algebra and al-

[7] https://universaldependencies.org/u/overview/morphology.html

gorithms as well as by National Excellence Programme 2018-1.2.1-NKP-00008: Exploring the Mathematical Foundations of Artificial Intelligence. A hardware grant from NVIDIA Corporation is gratefully acknowledged. GNU parallel was used to run experiments (Tange, 2011).

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of International Conference on Learning Representations*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136. Association for Computational Linguistics.

James R. Curran and Miles Osborne. 2002. A very very large corpus doesn't always yield reliable estimates.

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, and et. al. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of ISCA '17*.

András Kornai. 2012. Eliminating ditransitives. In Ph. de Groote and M-J Nederhof, editors, *Revised and Selected Papers from the 15th and 16th Formal Grammar Conferences*, LNCS 7395, pages 243–261. Springer.

András Kornai and Zsolt Tuza. 1992. Narrowness, pathwidth, and their application in natural language processing. *Discrete Applied Mathematics*, 36:87–92.

András Kornai, Attila Zséder, and Gábor Recski. 2013. Structure learning in weighted languages. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 72–82, Sofia, Bulgaria. Association for Computational Linguistics.

David J.C. MacKay. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

T.C. Mendenhall. 1887. The characteristic curves of composition. *Science*, 11:237–249.

Jason Merchant. 2001. *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*. Oxford University Press.

George A. Miller. 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63:81–97.

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

Csaba Oravecz, Tamás Váradi, and Bálint Sass. 2014. The Hungarian Gigaword Corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

David M.W. Powers. 1998. Applications and explanations of Zipf's law. In D.M. W. Powers, editor, *NEMLAP3/CONLL98: New methods in language processing and Computational natural language learning*, pages 151–160. ACL.

H.S. Sichel. 1974. On a distribution representing sentence length in written prose. *Journal of the Royal Statistical Society Series A*, 137(1):25–34.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. Srilm at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, volume 5.

O. Tange. 2011. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

W.C. Wake. 1957. Sentence-length distributions of Greek authors. *Journal of the Royal Statistical Society Series A*, 120:331–346.

C.B. Williams. 1944. A note on the statistical analysis of sentence-length as a criterion of literary style. *Biometrika*, 31:356–361.

G. Udny Yule. 1939. On sentence-length as a statistical characteristic of style in prose: with applicationsto two cases of disputed authorship. *Biometrika*, 30:363–390.

G. Udny Yule. 1944. *The Statistical Study of Literary Vocabulary*. Cambridge University Press.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley.

A Appendix

Theorem A.1. *Let us define f as $x = \frac{f(x)}{F(f(x))}$ with $F(0) > 0$, then*

$$\left[x^i\right](f(x))^k = \frac{k}{i}[x^{i-k}]F^i(x) \qquad (21)$$

Proof. By Lagrange–Bürmann formula with composition function $H(x) = x^k$. □

Theorem A.2. *In the Bayesian evidence if both the model and parameter a priori is uniform, then*

$$\mathbb{P}(\mathcal{H}_i \mid D) = \frac{\mathbb{P}(D \mid \mathcal{H}_i)\cdot\mathbb{P}(\mathcal{H}_i)}{\mathbb{P}(D)} \propto f(\mathbf{w}_i^*) +$$

$$\frac{1}{n}\cdot\ln\mathrm{Vol}(\mathcal{H}_i) + \frac{1}{2n}\ln\det f''(\mathbf{w}_i^*) + \frac{d}{2n}\cdot\ln\frac{n}{2\pi}$$

where $f(\mathbf{w}_i)$ is the cross entropy of the measured and the modeled distributions. See Equation 17.

If the augmented model (18) is used, then Equation 19 follows.

Proof.

$$\mathbb{P}(D \mid \mathcal{H}_i) \overset{\text{uniform a priori}}{=} \int \mathbb{P}(D \mid \mathbf{w}_i, \mathcal{H}_i) \cdot \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \, \mathrm{d}\mathbf{w}_i = \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot \int \prod_{x \in X} \mathbb{Q}_{\mathbf{w}_i}(x)^{n_x} \, \mathrm{d}\mathbf{w}_i =$$

$$\frac{\int \exp\left\{ -n \cdot \overbrace{\left(-\sum_{x \in X} \frac{n_x}{n} \cdot \ln \mathbb{Q}_{\mathbf{w}_i}(x) \right)}^{f(\mathbf{w}_i)} \right\} \mathrm{d}\mathbf{w}_i}{\mathrm{Vol}(\mathcal{H}_i)}$$

Using Laplace method:

$$\approx \frac{1}{\mathrm{Vol}(\mathcal{H}_i)} \cdot e^{-n \cdot f(\mathbf{w}_i^*)} \cdot \frac{\left(\frac{2\pi}{n}\right)^{\frac{d}{2}}}{\sqrt{\det f''(\mathbf{w}_i^*)}}$$

Taking $-\frac{1}{n}\ln(\bullet)$ for scaling (does not effect the relative order of the models):

$$\frac{1}{n} \ln \mathrm{Vol}(\mathcal{H}_i) + f(\mathbf{w}_i^*) + \frac{1}{2n} \ln \det f''(\mathbf{w}_i^*) + \frac{d}{2n} \cdot \ln\left(\frac{n}{2\pi}\right)$$

As for the augmented model, the model parameters are the concatenation of the original parameters and the auxiliary parameters. Thus the overall Hessian is the block-diagonal matrix of the original and the auxiliary Hessian. Similarly, the overall model volume is the product of the original and the auxiliary volume. Trivially, the logarithm of product is the sum of the logarithms.

Since the auxiliary model can fit the uncovered part perfectly: $p_x = (1 - \lambda) \cdot q_x$ on $x \notin \operatorname{supp} \mathcal{H}_i$. See (18) for that λ is the covered probability of the sample.

$$\begin{aligned} \mathbb{P}(D \mid \mathcal{H}_i') = &- \sum_{x \in X \setminus \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln p_x \\ &- \sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln\left(\lambda \cdot \mathbb{Q}_{\mathbf{w}_i^*}(x)\right) + \\ &\frac{1}{n} \cdot (\ln \mathrm{Vol}(\mathcal{H}_i) + \ln \mathrm{Vol}(\text{aux. model})) + \\ &\frac{1}{2n} \cdot \ln \det\left(\text{model Hessian}\right) + \\ &\frac{1}{2n} \cdot \ln \det\left(\text{aux. model Hessian}\right) + \\ &\frac{d'}{2n} \cdot \ln \frac{n}{2\pi} \end{aligned} \tag{22}$$

where d' is the overall parameter number.

Further, if one subtracts the entropy of the sample then only the first two term is changed compared to Equation 22 and Equation 19 follows.

$$\begin{aligned} &\sum_{x \in X} p_x \cdot \ln p_x - \sum_{x \in X \setminus \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln p_x \\ &- \sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln\left(\lambda \cdot \mathbb{Q}_{\mathbf{w}_i^*}(x)\right) = \\ &\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\lambda \cdot \mathbb{Q}_{\mathbf{w}_i^*}(x)} = \\ &\sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \left(\ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)} + \ln \frac{1}{\lambda} \right) = \\ &\lambda \cdot (-\ln \lambda) + \sum_{x \in X \cap \mathrm{supp}(\mathcal{H}_i)} p_x \cdot \ln \frac{p_x}{\mathbb{Q}_{\mathbf{w}_i^*}(x)} \end{aligned}$$

q.v. Definition 3.1. □