Some Remarks on Zipf's Law

Marcus Kracht, UCLA

Zipf's Laws make connections between three things: the length of a word, its probability and the probability rank. In this paper I shall be concerned with the connection between these in languages generated by a probabilistics context free grammar.

Recall that a probabilistics context free grammar is a context free grammar $G = \langle A, N, S, R, P_G\rangle$, where A is the alphabte of terminal symbols, N the alophabet of nonterminal symbols, $S \in N$ the start symbol, $R = \{\rho(i) : i < p\}$ the set of rules, and P_G a function which assigns probabilities to rules in such a way that the probabilities for rules that expand a given nonterminal sum to 1. Trees are in biunique correspondence with leftmost derivations. Probabilities for trees are assigned to their leftmost derivations as follows. Let $\vec{\rho} = \rho(i_1)\rho(i_2)\cdots\rho(i_n)$ be a derivation, then the probability of the derivation is $\prod_{j=1}^{n} P_G(\rho(i_j))$. The probability of a string is the sum of the probabilities for all its leftmost derivations.

The leftmost derivations can also be generated by a PCFG in the following way. For convenience we assume that the grammar rules are either of the form $X \to \vec{x}$, where $\vec{x}$ is a terminal string, or of the form $X \to \vec{Y}$, where $\vec{Y}$ is a string of nonterminals. Put $A^d := R$, $N^d := N$, and $S^d := S$, and let the new set of rules be

$$\begin{aligned} R^d := \quad & \{X \to \rho(i)\vec{Y} : i < p \text{ and } \rho_i = X \to \vec{Y}\} \\ & \cup \{X \to \rho(i) : i < p \text{ and } \rho(i) = X \to \vec{x}\} \end{aligned} \tag{1}$$

The rules of the new grammar are in biunique correspondence with the rules of the old grammar, and we assign the probabilities accordingly. The rule $P_{G^d}(X \to \rho(i)\vec{Y}) := P_G(\rho(i))$, and $P_{G^d}(X \to \rho_i) := P_G(\rho(i))$. The new grammar is $G^d = \langle A^d, N^d, S^d, R^d, P_{G^d}\rangle$. There is an obvious map from strings generated by G^d to strings generated by G. It is defined by induction on the derivation in G^d (which is again a different object than the strings of G^d). Since G^d produces strings in Polish Notation, one can also do induction on the string qua term:

$$\begin{aligned} \delta(X) &:= X \\ \delta(\rho) &:= \vec{x} \qquad (\rho = X \to \vec{x}) \\ \delta(\rho\vec{\sigma}_1\cdots\vec{\sigma}_n) &:= \delta(\vec{\sigma}_1)\cdots\delta(\vec{\sigma}_n) \qquad (\rho = X \to \vec{Y}) \end{aligned} \tag{2}$$

Of particular interest are the terminal strings of G^d. These correspond to the terminal strings of G in a biunique way, since the first clause of the definition above is not used.

Let A be an alphabet. The space $\mathbb{R}^A$ can be endowed with operations that turn it into a vector space with unit vectors a, $a \in A$. Similarly, we can form the set $\mathbb{N}^A$ and endow it with an operation of addition and linear multiplication. $\mathbb{N}^A \subseteq \mathbb{R}^A$. These spaces shall be endowed with the so–called **1–norm**: let $x = \sum_{a \in A} a x_a$. Then

$$(3) \qquad |v| := \sum_{a \in A} |x_a|$$

If $\vec{x}$ is a string, $|\pi(\vec{x})|$ is simply the length of $\vec{x}$. A subset S of $\mathbb{N}^A$ is called **linear** if there are vectors v, w_i, $i < p$, such that

$$(4) \qquad S = \left\{ v + \sum_{i=1}^{p} k_i w_i : \text{ for all } i < p : k_i \in \mathbb{N} \right\}$$

A finite union of linear sets is called a **semilinear set**. The **Parikh–map** π from A^* to $\mathbb{N}^A$ is given as follows.

$$(5) \qquad \begin{aligned} \pi(a) &:= a \\ \pi(\vec{x}{}^\frown\vec{y}) &:= \pi(\vec{x}) + \pi(\vec{y}) \end{aligned}$$

If $L \subseteq A^*$ is generated by a context free grammar then $\pi[L]$ is a semilinear set. (The converse is false.)

Now, the Parikh–map conflates many strings into the same vector. Therefore, let $f_L : \mathbb{N}^n \to \mathbb{N}$ be defined by $f_L(v) := |\{\vec{x} : \pi(\vec{x})|$. We say that the number $f_V(v)$ is the **population** of v. Let κ_L denote the number of strings of length $\leq n$ in L. We can derive the following formula.

$$(6) \qquad \kappa_n = \sum_{|v| \leq n} f_L(v)$$

We shall ask: how fast does this number grow? The upper limit is given by $\sum_{i \leq n} g^i = (g^{n+1} - 1)/(g - 1)$, $g := |A|$, thus an exponential growth. On the other hand, the sets of vectors of length n are bounded by a polynomial. Basically, we expect two types of languages: those that grow exponentially and those that grow polynimally.

Definition 1 *A language L is called **sparse** if κ_n is $O(n^q)$ for some q. This means that there is a polynomial f of degree q such that for almost all n, $\kappa_L \leq f(n)$.*

Theorem 2 *Let L be a context free language. Then if L is not sparse, there is α and a constant c such that for almost all n: $\kappa_n \geq c\alpha^n$.*

(Awaits proof.)

Now we shall look into the problem of the probabilities depending on rank. We shall make the assumption that there is an unambiguous probabilistic grammar for L. (If L is regular and accepted by a probabilities FSA, this is certainly the case.) Then the probabilty for a given string $\vec{x}$ is the probability of its unique leftmost derivation. This allows us to replace the original grammar by G^d.

The set of complete derivations is a semilinear subset of $\mathbb{R}^R$. Consider the following map from complete derivations to $\mathbb{R}$.

$$(7) \qquad g(\rho_1 \cdots \rho_n) := \sum_{i=1}^{n} -\log P_G(\rho_i)$$

(The base of the logarithm is unimportant.) Since probabilities are numbers strictly between 0 and 1, $g(\vec{\rho}) > 0$. $g(\vec{\rho})$ is actually the negative logarithm of the probability of the derivation, since the probabilities multiply, so that the logarithms add. Moreover, notice that g factors through the Parikh–map. Namely, we can define the following map from $\mathbb{R}^R$ to $\mathbb{R}$:

$$(8) \qquad h(k_1\rho(1) + \cdots k_n\rho(n)) := \sum_{i=1}^{n} -k_i \log P_G(\rho(i))$$

Then $g(\vec{\rho}) = h(\pi(\vec{\rho}))$, because the probability is independent of the order in which the rules appear.

Now consider Zipf's Second Law. A **probabilistic language** is a pair $\langle L, p\rangle$ such that $\sum_{\vec{x}\in L} p(\vec{x} = 1$. Suppose the language is infinite, and that L is enumerated by a function r such that $pr(k)) \leq p(r(m))$ whenever $k > m$. Such a function is called a **rank function**. Then the law asserts that $p(r(m)) \approx c \cdot m^{-\alpha}$, where $\alpha > 1$. Thus, if the strings are aligned in decreasing probability, the probabilities go down in a polynomial fashion.

Now let us establish a rank function for the language L generated by an unambiguous PCFG G. We have established a map from $\pi(\vec{x})$ to the negative logarithm of the probabilitiy. Now, if $r < s$ then $-\log r > -\log s$, so in order to order the elements according to decreasing rank, we may also order them with increasing negative logarithm of probability. Let $o(x) = |\{v \in \mathbb{N}^R : h(v) \leq x\}|$. The numbers $o(x)$ grow polynomially with x (they measure the volume of the preimage of the cube of vectors of length $\leq x$; the latter is polynomial in x, and h is a linear map). Now, what we actually want to know is another number, namely

$$(9) \qquad p(x) := |\{\vec{\rho} \in L(G^d) : -\log P_{G^d}(\vec{\rho}) \leq x\}|$$

We distinguish two cases. First, assume that L is sparse. Then $p(x)/o(x)$ is a polynomial. Let then b be a given rank. Then x is of the magnitude $\gamma b^{1/w}$ for some numbers γ and w. So, the probability is $e^{-\gamma b^{1/w}}$. Thus, the probabilities decrease not in polynomial fashion, but in an exponential fashion (though the function is more exactly an exponential of the wth root of the rank b).

Now let us assume that L is not sparse; then $p(x)/o(x)$ is exponential. In this case, if b is the rank, x is of the magnitude $\gamma + \delta \log b$ for some constants γ and δ. So, if $-\log p(r(b)) = x \approx \gamma + \delta \log b$, we have $p(r(b)) = e^{-\gamma-\delta \log b} = e^{-\gamma} \cdot e^{-\delta \log b}$. Put $\vartheta := e^{-\gamma}$; furthermore, observe that $e^{-\delta \log b} = (e^{\log b})^{-\delta} = b^{-\delta}$. So we arrive at the following theorem.

Theorem 3 *Let $\langle L, p \rangle$ be a probabilistic language generated by an unambiguous PCFG. Suppose further that L is not sparse. Then the function from rank b to probability is given in the limit by $p(b) = \vartheta b^{-\delta}$ with $\vartheta > 0$, $\delta > 1$.*