Here, df_1 is the same as standard document frequency (df). df_2 plays an important role in *adaptation*, a term borrowed from the literature on language modeling for speech recognition [13, chapter 14], where there is considerable interest in adapting the probabilities to the first few words of a document. If "Noriega" is mentioned early in the document, chances are that that word (and its friends) will be mentioned again[15]. Psychologists use the term priming [1] to reflect the fact that people react quicker and more accurately to "nurse" if it has been primed by a highly associated word like "doctor."

We define adaptation to be the chance that a term will be mentioned again, given that we have seen it before.

Definition 2. $adaptation \equiv Pr(k \geq 2 | k \geq 1) \approx df_2/df_1$

There are huge quantity discounts, especially for good keywords. The first mention costs $-log(df_1/D)$ bits, but subsequent mentions are cheaper: $-log(df_2/df_1)$ bits. For good keywords like "Noriega," the first mention is quite surprising (e.g., "Noriega" is mentioned in one document in a thousand Associcated Press (AP) stories, shortly after the US invasion of Panama), but the subsequent mention is less so (more than half of the documents that mention "Noriega" once, mention him a second time). There is considerably less adaptation for function words and meaningless random substrings, where the first mention and subsequent mentions are about equally likely (no quantity discounts).

This discounting view is reminiscent of the Given-New Distinction in Discourse Theory [3], which is commonly used in intonation studies such as [4]. The first mention ("new" information) is marked, whereas subsequent ("given") mentions are unmarked. In statistical terms, first mentions tend to be more surprising than subsequent mentions, at least for meaningful words. Random substrings behave more randomly, with less difference between first mentions and subsequent mentions.

In Japanese, we find that words adapt more than most substrings of Japanese characters, and therefore we believe adaptation could be useful in word-breaking applications for Asian languages.

2 Suffix Arrays

A suffix array [9] is a convenient data structure for computing the frequency and location of a substring (ngram) in a large corpus. The input text (corpus) is a sequence of N tokens. Tokens can be words, bytes, Asian characters, etc.

We will use different tokenization rules from time to time. The simplest tokenization rule is to split the text up into bytes, starting a suffix at each byte position. In this case, the number of suffixes, S, is the same as the number of bytes in the input corpus N. For Japanese and Chinese text, it is more appropriate to tokenize by characters (typically 2-bytes), rather than by bytes, so that $S \approx N/2$. For English text, it is often convenient to start suffixes at word boundaries so $S \approx N/5$. The code posted at [21] provides options for different tokenization

16. Daniel Jurafsky and James H. Martin: Speech and Language Processing. Prentice Hall, Upper Saddle River, NJ (2000)
17. Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon: Spoken Language Processing, Prentice Hall, Upper Saddle River, NJ (2001)
18. R. Harald Baayen: Word Frequency Distributions, Kluwer Academic Publishers, Dordrecht, The Netherlands (2001)
19. Mikio Yamamoto and Kenneth Church: Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus, Computational Linguistics, vol. 27, No. 1, 1–30 (2001)
20. Yinghui Xu and Kyoji Umemura: Improvements of Katz K Mixture Model, Information and Media Technologies, vol. 1, no. 1, 411–435, (2006)
21. Kyoji Umemura: www.cicling.org/2009/Umemura-Church/