Statistical Techniques for Language Recognition: An Empirical Study Using Real and Simulated English

Ravi Ganesan
Research and Strategic Planning
Bell Atlantic
Beltsville, Maryland 20705

Alan T. Sherman*
Computer Science Department
University of Maryland Baltimore County
Baltimore, Maryland 21228-5398

September 28, 1993

Abstract

Computer experiments compare the effectiveness of five test statistics at recognizing and distinguishing several types of real and simulated English strings. These experiments measure the statistical power and robustness of the test statistics X^2, *ML*, *IND*, S, and *IC* when applied to samples of everyday American English from the Brown Corpus and *Wall Street Journal* and to simulated English generated from 1st-order Markov models based on these samples. Here, X^2 is the chi-squared test statistic; *ML* is a likelihood ratio test for recognizing a known language; *IND* is a likelihood ratio test for distinguishing unknown 0th-order noise from unknown 1st-order language; S is a log-likelihood function that is a most-powerful test for distinguishing a known language from uniform noise; and *IC* is the index of coincidence. The test languages comprise four types of real English, two types of simulated 1st-order English, and three types of noise.

Two experiments characterize the distributions of these test statistics when applied to nine test languages, presented as strings of different lengths and contaminated with various amounts of noise. Experiment 1 varies the length of the string from 2 to 2^{17} characters. Experiment 2 adds uniform noise to samples of three fixed lengths (2^4, 2^7, 2^{10}), with the amount of added noise ranging from 0% to 100%. These experiments assess the performance of the test statistics under realistic cryptographic constraints. An empirical approach is needed because the asymptotic theory of statistical inference on Markov chains does not apply to short strings drawn from natural language.

Through graphs and tables of observed statistical power, we compare the effectiveness of the test statistics at distinguishing various pairs of languages at several critical levels. Although no statistic dominated all others for all critical levels and string lengths, each test performed well at its designated task. For distinguishing a known type of English from uniform noise at critical levels 0.1 through 0.0001, X^2 attained the highest power, with *ML* and S also performing well. For distinguishing uniform noise from a known type of English at the same critical levels, *ML* had the overall best performance, with *IC*, X^2, S, and *IND* also performing well. And for each of these tasks under noisy conditions, *ML* attained the highest power. In addition, through histograms we describe the actual distribution of each statistic on various language types. These detailed results, which show relationships between power, critical level, and string length, will help cryptanalysts and others apply statistical methods to practical language-recognition problems.

*Part of this work was carried out while Sherman was a member of the Institute for Advanced Computer Studies, University of Maryland College Park.

Keywords. Automatic plaintext recognition, chi-squared test statistic, computational linguistics, cryptanalysis, cryptography, cryptology, document processing, index of coincidence, language recognition, likelihood ratio tests, Markov models of language, maximum likelihood estimators, natural language processing, statistical inference, statistical pattern recognition, statistics of language.

1 Introduction

Automatic language recognition plays an important role in cryptanalysis, speaker identification, document processing, and other pattern-recognition tasks. To carry out such language-recognition tasks, the theory of statistics offers useful models and test statistics.[1] For example, in our companion introductory guide [16], we explain how to solve four well-defined language-recognition problems using likelihood ratio tests and other standard statistical techniques, when languages are modeled as Markov chains. Although the theory of statistical inference has much to say about the asymptotic performance of many test statistics when applied to the idealized languages of Markov models, this theory says little, if anything, about the performance of test statistics on short strings drawn from natural languages such as English. Yet most practitioners must deal with such input strings.

This paper presents and analyzes results of computer experiments that characterize the distributions of five test statistics when applied to several types of real and simulated English strings of various lengths. We focus on statistical approaches to language recognition and empirically analyze the performance of five test statistics for recognizing and distinguishing nine test languages under realistic practical constraints. Specifically, we answer the following questions. What is the actual distribution of each statistic when applied to several types of real English? How do these distributions vary with the length of the input string? For each test statistic, how does its distribution on real English compare with that on simulated English generated from a 1st-order Markov model? How are these distributions affected when the input is contaminated with noise? How effective is each test statistic at distinguishing various pairs of languages? And what helpful advice can be offered to practitioners who wish to solve language-recognition problems?

To answer these questions we perform two experiments: Experiment 1 characterizes the distributions of five test statistics when applied to strings from nine types of language, with strings ranging in length from 2 to 2^{17} characters. Experiment 2 characterizes the distributions of these statistics when applied to strings of three fixed lengths (2^4, 2^7, 2^{10}) from the same languages, when the strings are contaminated with various amounts of added uniform noise. To compare the effectiveness of the test statistics at distinguishing various pairs of languages, we compute the power of these statistics at several critical levels.

Each experiment studies the five test statistics X^2, ML, IND, S, IC, and normalized variations thereof, defined in Section 2. We apply these test statistics to strings randomly selected from large samples of text drawn from nine test languages. Our nine test languages comprise four types of real English, two corresponding 1st-order Markov models of English, and three alternatives to English consisting of uniform noise, a non-uniform 0th-order noise, and a simple repeating pattern. Parameters for each Markov model, including the two 1st-order models of English and the non-uniform 0th-order noise, are computed from two of the samples of real English. We drew our four samples of real English from the *Wall Street Journal* and from the Brown Corpus, which is a well-known collection of everyday American English, assembled during 1963–1964 at Brown University under the direction of Francis, and analyzed by Kučera and his colleagues [13, 26].

Much is known about the asymptotic theory of statistic inference on Markov chains. For example, Anderson and Goodman [2]; Billingsly [6, 7]; and Kullback, Kupperman, and Ku [29] survey the literature. In addition, a few experiments have been carried out to determine the exact and approximate distributions of some our test statistics when applied to theoretical languages

[1]We assume the reader is familiar with elementary statistics—as explained by Larsen and Marx [30], for example. For a more advanced review of hypothesis testing, see Lehmann [31].

defined by certain multinomial distributions. For example, see Good, Gover, and Mitchel [24]; Good and Crook [10, 11, 23]; Good [21]; and West and Kempthorne [38]. Also, Davies and Ganesan [12] give histograms of their normalization of the S statistic applied to eight-letter passwords, and there are several experimental studies on estimating model parameters (for example, see Church and Gale [9] and other works cited in Section 4 of our companion guide). And developing new tests of nonrandomness remains an active area of research—for example, see Schrift and Shamir [35]. But we are not aware of any previous experimental work that characterizes the distributions or robustness of our test statistics applied to natural language. Our work helps fill this gap in the understanding of statistical techniques for language recognition.

The rest of this paper is organized as follows. Section 2 defines the five test statistics studied in our experiments. Section 3 explains each of our two experiments, including our procedures, toolkit, and test languages. Section 4 presents and analyzes the main results of our experiments. Section 5 summarizes our conclusions. In addition, Appendix B lists selected detailed results of Experiment 1 in tabular form. Additional detailed experimental data in graphs and tables are given in the six supplemental appendices of our technical report [17]. For the reader unfamiliar with our introductory guide [16], Appendix A briefly reviews our approach to language recognition.

2 Five Test Statistics

Our experiments evaluate the five test statistics X^2, ML, IND, S, and IC, defined in this section. Here, X^2 is the chi-squared test statistic; ML is a likelihood ratio test for recognizing a single known language; IND is a likelihood ratio test for distinguishing unknown 0th-order noise from unknown 1st-order language; and S is a log-likelihood statistic that is asymptotically equivalent to a most powerful test for distinguishing a known language from uniform noise.[2] Finally, IC is Friedman's [14] index of coincidence, which Good, Gover, and Mitchell [24] claim is useful for detecting non-uniform unknown language in some cases where the X^2 statistic does not apply. Each of these statistics compares the features of a candidate string with those of a known model, where features are the k-gram frequency counts for some k.[3]

For simplicity, and since higher-order grams are not available in all applications, we state all formulas and carry out all experimental work with $k = 2$; that is, we work within 1st-order (bigram) models. Our methods, however, apply for any k, and the practitioner will likely obtain best statistical results by using the highest reasonable value of k.

To express the test statistics on a convenient scale we also work with normalized forms, defined in this section. Specifically, for X^2, ML, and IND, we work primarily with the corresponding normalized test statistics $\diamond X^2$, $\diamond ML$, and $\diamond IND$, which are based on asymptotic chi-squared interpretations. For the test statistics S and IC, we work both with their raw values and with their equivalent normalized forms $\hat{S}$ and $\hat{IC}_\star$. In addition, we compute Anderson's [1] variation of S, which variation we denote by A.[4]

For each of our four types of real English, we estimated transition probabilities from a base sample of over 500,000 characters, using straightforward maximum-likelihood estimates as described

[2] In our companion guide [16] we refer to S as Sinkov's [36] statistic, but other cryptanalysts—including Friedman [15], Callimahos and Friedman [8], and Kullback [28]—have also advocated this natural statistic.

[3] We use the term *k-gram* to refer to any sequence of exactly k letters. For $k = 1, 2, 3$, we call any such *gram* a *unigram*, *bigram*, or *trigram*, respectively.

[4] In our companion paper we denote the test statistics S and A by S_2 and A_2, respectively.

in Section 3.2. Throughout we work with Markov models with $m = 27$ states corresponding to the characters 'A'–'Z' and blank.

Notation

Each of the test statistics X^2, ML, and S is defined in terms of the transition probabilities $[p_{ij}]_{1\leq i,j\leq m}$ of the base language and the observed bigram frequency counts $\{n_{ij}\}_{1\leq i,j\leq m}$ of the candidate string; the other statistics IND and IC are defined from these frequency counts alone. The candidate string of n characters forms $N = n - 1$ overlapping bigrams. For each $1 \leq i, j \leq m$, let n_i denote the observed frequency of letter i. There are $n_{i*} = \sum_{j=1}^{m} n_{ij}$ bigrams beginning with the letter i. For all $1 \leq i \leq m$, either $n_i = n_{i*}$ or $n_i = n_{i*} + 1$.

From these frequency counts, define the *observed relative unigram frequencies* $\hat{p}_i = n_i/n$ and the *observed relative transition frequencies* $\hat{p}_{ij} = n_{ij}/n_{i*}$. In addition, let $b_{ij} = p_i p_{ij}$ denote the *unconditional bigram probability* of the bigram ij in the Markov model of the base language, where p_i denotes the steady-state state probability (*i.e.* unigram probability) from the Markov model of the base language.

Test Statistics X^2, ML, IND and Normalized Variations $\diamond X^2$, $\diamond ML$, $\diamond IND$

The asymptotically equivalent test statistics X^2 and ML are defined by

$$X^2 = \sum_{1\leq i,j\leq m} \frac{(n_i\hat{p}_{ij} - n_i p_{ij})^2}{n_i p_{ij}} \tag{1}$$

and[5]

$$ML = 2 \sum_{1\leq i,j\leq m} n_{ij} \ln \frac{\hat{p}_{ij}}{p_{ij}}. \tag{2}$$

When the base and candidate languages are generated by the same Markov chain, X^2 and ML have an asymptotically χ^2 distribution with $\nu_1 = m(m-1) - d$ degrees of freedom, where d is the number of zero transition probabilities from the base language. In each experiment we took the base language to be language BCa (see Section 3.1), for which $d = 168$ and thus $\nu_1 = 27(27-1) - 168 = 534$. In Equations 1, 2 and throughout, we adopt the convention that all summations are computed over only those indices i, j such that $p_{ij} \neq 0$.[6]

Similarly, when the candidate language is 0th order, the likelihood ratio test statistic

$$IND = 2 \sum_{1\leq i,j\leq m} n_{ij} \ln \frac{\hat{p}_{ij}}{\hat{p}_j} \tag{3}$$

has an asymptotically χ^2 distribution with $\nu_2 = (m-1)^2 - d$ degrees of freedom. For our base language BCa, $d = 168$ and thus $\nu_2 = (27-1)^2 - 168 = 508$. Supplemental Appendix C [17] lists the values of d, ν_1, and ν_2 for each of the other test languages. For example, the value of d for the English test languages BCf, BCg, and WSJ1 is 185, 163, and 141, respectively.

---

[5] Throughout let $\ln = \log_e$ and $\lg = \log_2$.

[6] For more elaborate ways of treating zero transition probabilities, see Good [19], Church and Gale [9], Davies and Ganesan [12], and Levin and Reeds [33].

To express the test statistics X^2, ML, IND on a convenient scale, we normalize them with a transformation suggested by Trevedi [37] to produce the corresponding normalized test statistics $\diamond X^2$, $\diamond ML$, and $\diamond IND$. Specifically, for any asymptotically χ^2 test statistic ψ with $\nu > 100$ degrees of freedom, we compute

$$\diamond\psi = \sqrt{2\psi} - \sqrt{2\nu - 1}, \tag{4}$$

which has an approximately standard normal distribution.[7] Thus for base language BCa, $\diamond X^2 = \sqrt{2X^2} - \sqrt{1067}$, $\diamond ML = \sqrt{2ML} - \sqrt{1067}$, and $\diamond IND = \sqrt{2IND} - \sqrt{1015}$.

Each of our statistics is computationally simple in the sense that it can be easily computed in time $\Theta(n)$, after precomputing the transition probabilities for the base language. Nevertheless, for applications that demand extreme performance, a variety of minor refinements can be implemented to reduce the constant factors in the computation time. For example, Knuth [25] describes an alternate way to compute the X^2 statistic. And to compute ML, IND, and S, some time can be saved by precomputing the logarithm of each transition probability.

The S Statistic and Variations $\hat{S}$ and A

The S statistic for distinguishing known language from uniform noise is the log-likelihood function

$$S = \sum_{1 \le i,j \le m} n_{ij} \ln p_{ij}, \tag{5}$$

which is asymptotically equivalent to a most powerful test for the same task based on the Neyman-Pearson Lemma. To express S on a convenient scale, we center and normalize S/N by

$$\hat{S} = \frac{(S/N) - \mu_S}{\sigma_S/\sqrt{N}}, \tag{6}$$

as did Baldwin and Sherman [3]. When the candidate sample consists of independent bigrams from the base language, the test statistic $\hat{S}$ has an asymptotically standard normal distribution.

In Equation 6, the constants μ_S and $\sigma_S = \sqrt{\sigma_S^2}$ are computed from the base language parameters by

$$\mu_S = \sum_{1 \le i,j \le m} b_{ij} \ln p_{ij} \qquad \text{and} \qquad \sigma_S^2 = \sum_{1 \le i,j \le m} b_{ij}((\ln p_{ij}) - \mu_S)^2, \tag{7}$$

where p_{ij} and b_{ij} are, respectively, the conditional and unconditional bigram probabilities of the bigram ij in the base language. In each experiment we used the constants $\mu_S = -2.355419$ and $\sigma_S = 1.043849$, as computed by Equation 7 for base language BCa. For interest and comparison, supplemental Appendix C lists the base language constants μ_S and σ_S for each of our nine test languages. For example, the values of $\mu_S \pm \sigma_S$ for the English test languages BCf, BCg, and WSJ1 are approximately -2.328 ± 1.055, -2.325 ± 1.058, and -2.370 ± 1.030, respectively.

Anderson's variation of S for bigrams is

$$A = \prod_{1 \le i,j \le m} b_{ij}^{n_{ij}(n/(2N))}. \tag{8}$$

For convenience, we worked with the equivalent statistic $\ln A$.

[7] A *standard normal* distribution is a Gaussian distribution with mean 0 and variance 1.

The Index of Coincidence IC and Normalized Variation $\hat{IC}_\star$

The index of coincidence IC for bigrams is defined by

$$IC = \sum_{1 \le i,j \le m} \frac{n_{ij}(n_{ij}-1)}{N(N-1)}. \tag{9}$$

Good and his colleagues [23, 24] present experimental evidence that, when applied to uniform noise and when $N \ll m^2/12$ (for us, $m^2/12 = 27^2/12 \approx 61$), the statistic $IC_\star = N(N-1)IC/2$ has an approximately Poisson distribution with mean $\mu_\star = N(N-1)/(2m^2)$. For more about the IC and its distribution, see Good [22, 20, 21] and Kullback [27, pp. 151–153].

To express IC on a convenient scale, we center and normalize $IC_\star$ by

$$\hat{IC}_\star = \frac{IC_\star - \mu_\star}{\sqrt{\mu_\star}}. \tag{10}$$

When the test statistic $IC_\star$ is approximately Poisson, and when $\mu_\star$ is sufficiently large (*e.g.* $\mu_\star \ge 5$), the related test statistic $\hat{IC}_\star$ is approximately standard normal. However, since the condition $\mu_\star \ge 5$ is satisfied only when $N \ge 86$, we do not recommend using a standard normal interpretation of $\hat{IC}_\star$ for short strings. To prevent arithmetic overflow when computing IC, it is helpful not to distribute the factor $1/(N(N-1))$ outside of the summation.

3 Experimental Methods

We perform two experiments to determine how each of the test statistics X^2, ML, IND, S, and IC performs when applied to real and simulated English under realistic practical constraints. This section describes our two experiments in detail, focusing on our methods. In doing so, we explain our experimental toolkit, our test languages, how we preprocessed our base samples of these languages, how we carried out each experiment, and how we analyzed the results.

3.1 The Nine Test Languages

To carry out a run of any experiment on any test statistic, up to two languages must be specified: a *candidate language* from which candidate strings are drawn, and a *base language* (if needed) whose known transition probabilities are used in some test statistics. We ran our experiments using the nine test languages described in Table 1, each of which is defined by a sample file of over 500,000 characters. These nine test languages comprise four types of everyday American English (BCa, BCf, BCg, WSJ1), two types of simulated English (1st-order BCa, 1st-order BCf), and three types of noise (0th-order BCa, uniform noise, "er" repeated).

Languages BCa, BCf, BCg, and WSJ1 are four examples of real English. As explained in Table 1, each of the languages BCa, BCf, and BCg is defined by a file in the Brown Corpus. BCa consists of general press reportage; BCf consists of popular lore; and BCg consists of Belles lettres, biographies, and essays. To learn more about these files and the Brown Corpus, see Francis and Kučera [13, 26]. Language WSJ1 is defined by a file of general newspaper articles from the *Wall Street Journal.* For test statistics that require a known base language, for simplicity, we always used BCa as the base language.

The simulated languages 1st-order BCa and 1st-order BCf are 1st-order Markov models of BCa and BCf, respectively, whose transition probabilities were computed from the base samples for BCa and BCf. Using our Markov chain simulator, we generated a large sample for 1st-order BCa and for 1st-order BCf. In this simulator, to select the initial state, we use the steady-state state (*i.e.* unigram) probabilities of the model. These two simulated languages allow us to compare the performance of test statistics when applied to real and idealized language.

The languages 0th-order BCa, uniform noise, and "er" repeated provide three alternatives to valid English. *0th-order BCa* is a 0th-order Markov model of BCa, whose non-uniform unigram probabilities were computed from the base sample for BCa. *Uniform noise* is a 0th-order Markov chain with uniform unigram probabilities. Using our Markov chain simulator, we generated a large sample for 0th-order BCa and for uniform noise. Finally, *"er" repeated* is simply a long string "ererer..." consisting of the overlapping bigrams "er" and "re", without any spaces. 0th-order BCa is useful in studying statistics that distinguish unknown 0th-order noise from unknown 1st-order language, and uniform noise is a convenient and practical model of invalid language. We experimented with the language "er" repeated as an extreme example of structure and as a debugging check.

The sample initial strings listed in Table 1 show some interesting similarities and differences among the test languages. For example, consider the word lengths in each sample. As expected, 0th-order BCa and uniform noise have much longer words on average than do BCa and 1st-order BCa. The expected word length in 0th-order BCa is $1/p_{27} \approx 5.72$, where $p_{27} = 88,971/508,567 \approx 0.1749$ is the unigram probability of blank in BCa English. Similarly, the expected word length in uniform noise is 27 characters. By contrast, according to Kučera and Francis [26, p. 366], the expected number of characters per word in the Brown Corpus is approximately 4.74 with standard deviation 2.66. Although the word lengths in BCa English and 1st-order BCa are roughly similar, 1st-order BCa and 1st-order BCf tend to have slightly longer words, reflecting the fact that BCa English is not a 1st-order Markov chain.

At first we were surprised to observe the striking similarity between the initial strings of 1st-order BCa and 1st-order BCf. We believe this similarity is due to the similarity of BCa and BCf transition probabilities and the fact that we generated these two simulated languages using the same pseudorandom number generator started from the same initial seed.

Despite the statistical appropriateness of the bigram frequencies in our computer generated samples of 1st-order BCa and 1st-order BCf, most humans would not classify these strings as English. In this sense, our 1st-order model of language coarsely models human language.

In selecting test languages for real English, we sought well-defined types of everyday English for which we could easily obtain a large sample in ASCII form. We worked with the Brown Corpus and the *Wall Street Journal* since the Maryland Intelligent Text Laboratory at the University of Maryland Baltimore County maintains a database of these and other samples of natural language. To measure how broadly our test statistics recognize language, we experimented with several different types of candidate English. But subject to this constraint, to minimize our work we kept the total number of test languages as small as possible. We ended up with the aforementioned nine test languages.

Table 1: The nine test languages. Each test statistic is applied to randomly generated strings from a sample file for each of these test languages. These nine languages consist of four types of American English (BCa, BCf, BCg, WSJ1), two types of simulated English (1st-order BCa, 1st-order BCf), and three types of noise (0th-order BCa, uniform noise, "er" repeated). The simulated languages 1st-order BCa and 1st-order BCf are 1st-order Markov models of BCa and BCf, respectively. For test statistics that require a known base language, we always used BCa as the base language.

Language Type	Source	Description	Length (chars)	Sample
English				
BCa	Brown Corpus file a0144	press reportage	508,567	THE FULTON COUNTY GRAND JURY SAID FRIDAY AN INVESTIGATION OF
BCf	Brown Corpus file f0148	popular lore	556,753	IN AMERICAN ROMANCE ALMOST NOTHING RATES HIGHER THAN WHAT THE
BCg	Brown Corpus file g0175	Belles lettres, biographies, essays	874,601	NORTHERN LIBERALS ARE THE CHIEF SUPPORTERS OF CIVIL RIGHTS AN
WSJ1	*Wall Street Journal* file wsj1	press reportage	763,770	PIERRE VINKEN YEARS OLD WILL JOIN THE BOARD AS A NONEXECUTIVE
Simulated English				
1st-order BCa	computer generated	1st-order Markov model using BCa transition probabilities	550,002	AN H QUND CHONEND S TISTILOFOLART TOR BUTHEY KATE OFIMIS BUDO
1st-order BCf	computer generated	1st-order Markov model using BCf transition probabilities	550,005	AN H RAPE CHOMEND S UPRKIMOFREART URO BUTHE PLATHORAMES PE BE
Not English				
0th-order BCa	computer generated	0th-order Markov model using BCa transition probabilities	550,002	AN I SAOE EJVHGIDTTW OTNNHRBNEAPT RR EW IESRNATETPBGIOTRDYDI
uniform noise	computer generated	0th-order Markov model using uniform transition probabilities	600,001	BMYJZRANFZFKVHHJDSTVYNUMMIQBNF BPSZYRQZDVXJFSRMAUEUPBHIOUQDWDJ
er repeated	computer generated	"ererer..."	628,860	ERERERERERERERERERERERERERERERERERERERERERERERERERERERERERERER

3.2 Preprocessing the Test-Language Samples

Before running the experiments, we preprocessed each of the nine test-language samples in two ways. First, we stripped each sample of extraneous characters. This preprocessing step converts each raw sample into a clean and convenient form consisting only of the upper-case letters 'A'–'Z' and blank. All candidate strings were drawn from clean samples. Second, we estimated unigram and transition probabilities for each clean sample, so that we could select any of the nine test languages as the base language.

The stripping process cleans up each sample as follows. 1) Each lower-case letter ('a'–'z') is converted to upper case. 2) Each non-alphabetic character (including <carriage-return> and all other special characters) is replaced by one blank. 3) Each sequence of two or more blanks is condensed into one blank. In Table 1, the file lengths refer to the lengths of the clean samples.

To estimate the transition probabilities from a clean sample, we computed straightforward maximum-likelihood estimates, as described for example by Good [19] and Bhat [5]. Thus, for each bigram ij, we estimated the transition probability p_{ij} by $\hat{p}_{ij} = n_{ij}/n_{i*}$, where n_{ij} is the observed frequency of bigram ij, and n_{i*} is the number of observed bigrams beginning with letter i. Similarly, for each unigram i, we estimated the unigram probability p_i by $\hat{p}_i = n_i/n$, where n_i is the observed frequency of unigram i, and n is the number of unigrams in the sample. For convenience, we set up a directory of language parameters, storing each matrix of transition probabilities and each array of unigram probabilities in a separate file.

As a result of the stripping process, our blank character gains two special properties. First, the stripping process can create more blank characters than appear in the raw sample; thus, our blank character differs slightly from the usual blank character in English. Second, the bigram blank-blank never appears in any base sample nor in any unadulterated candidate string. Although it is possible for the blank-blank bigram to appear in noisy candidate strings and in 0th-order BCa, when it does we ignore it. Thus, the bigram blank-blank never affects any calculation directly. By contrast, bigrams that contain exactly one blank are included in all calculations.

3.3 Experiment 1: Effect of Message Length on Tests

Experiment 1 empirically characterizes the distribution of each of our five test statistics when applied to strings of various lengths drawn from each of our nine test languages. The purpose of this experiment is to characterize the actual distributions of these five test statistics when applied to short and moderate length strings drawn from everyday English. Such "non-asymptotic" string lengths are typical of inputs encountered in many applications; the actual distributions are needed in order to interpret the values of the test statistics. By sampling strings of a wide range of lengths, Experiment 1 also shows at what lengths (if any) the theoretical asymptotic distributions of the test statistics accurately model their actual distributions on our test languages. Furthermore, by including real and simulated English in our test languages, for each statistic we can observe subtle differences that may arise in its actual distributions on real and simulated language, caused by (minor) differences between real English and our 1st-order model of it.

To carry out one run of Experiment 1, the experimenter must specify the following inputs: a list of test statistics, a base language, and a candidate language. Throughout we used the test statistics $\diamond X^2$, $\diamond ML$, $\diamond IND$, S, IC, and $\ln A$, as defined by the display equations in Section 2. As debugging checks and to study the distributions of S and $\hat{IC}_\star$, we also computed S/N, $\hat{S}$, and $\hat{IC}_\star$. For simplicity, we always chose the base language to be BCa. We chose the candidate language from the nine test languages described in Section 3.1. More specifically, as explained in Section 3.2, the candidate language was given by one of the nine clean language samples, and the base language was given by a file of transition probabilities estimated from the clean BCa sample. Each run of Experiment 1 applies all specified test statistics to the same strings generated from the given candidate language.

We apply one test statistic to one candidate language as follows. For each integer $1 \leq e \leq 17$, we select $h = 100$ candidate sample strings of length $n = 2^e$ characters ($N = n - 1$ bigrams) from the candidate language file. Thus, the candidate strings range in length from 2 to $2^{17} = 131,072$ characters. Each string is selected independently, with uniform distribution from the set of all possible substrings of length n from the specified language file. The test statistic is computed on each of the h strings. To summarize the distribution of the test statistic on the specified language, for each string length, Experiment 1 outputs the sample mean and sample standard deviation computed from the observed h values of the test statistic.[8] Optionally, when we desired a more detailed characterization of the distribution of the test statistic, we output a list of these values and viewed them as a histogram.

We arbitrarily chose the values 100 and 17 with the intent of achieving a sufficiently large sample size and of ensuring that asymptotic behavior of the test statistics would be apparent by our maximum string lengths. After performing a complete set of experiments with $h = 100$, we ran a smaller set with $h = 10,000$ and $1 \leq \lg n \leq 11$ for the purpose of improving the accuracy of our power calculations. All power calculations for Experiments 1 and 2 are based on sample size $h = 10,000$.

It is possible for the sample strings to overlap. Overlap is unlikely for short strings; substantial overlap is likely for huge strings.

To prevent potential run-time arithmetic overflow errors in calculating any test statistic, we set a "zero-cutoff value" $0 < \epsilon \ll 0.1$. All probabilities and relative frequencies less than ϵ were treated as zeroes. Thus, viewing each test statistic as a sum involving probabilities and relative frequencies, we computed each test statistic over only those summands that have probabilities and relative frequencies at least ϵ. Throughout we arbitrarily used $\epsilon = 10^{-6}$. Although this minor feature might be needed for some base samples, none of the nonzero transition probabilities for any of our test languages fell below the threshold.

3.4 Experiment 2: Noisy Plaintext

Experiment 2 is a variation of Experiment 1 that empirically characterizes the distribution of each of our five test statistics, when applied to strings contaminated with various amounts of added noise. As with Experiment 1, we apply each test statistic to sample strings drawn from each of our nine test languages. Unlike Experiment 1, however, for simplicity we work only with strings of three fixed lengths: $2^4 = 16$, $2^7 = 126$, $2^{10} = 1,024$. The purpose of this experiment is to characterize the actual distributions of our test statistics when applied to everyday English under noisy conditions, which are common in practical language-recognition applications.

Input to Experiment 2 consists of a list of test statistics, a base language, and a candidate language. For each test statistic, Experiment 2 outputs the sample mean and sample standard deviation of $h = 100$ values of the test statistic computed for each string length at each of several noise levels. We tried all noise levels $0 \leq w \leq 100$ in increments of 5. As with Experiment 1, we

[8] Specifically, we computed sample standard deviation as the square root of the unbiased sample variance.

worked with the test statistics $\diamond X^2$, $\diamond ML$, $\diamond IND$, S, IC, and $\ln A$, and we always chose the base language to be BCa.

For each string length n and noise level $0 \leq w \leq 100$, Experiment 2 selects h strings at random from the given candidate language file. Each character of each string is processed independently, and with probability $w/100$ replaced with a randomly chosen character (possibly the same character). The replacement character is chosen independently with uniform distribution over our 27-character alphabet. Each specified test statistic is applied to the same strings.

3.5 Computing Statistical Power

To compare our test statistics, we compute their observed statistical power at distinguishing various pairs of languages at several critical levels.[9] In this section we explain how we approximate statistical power from our observed means and standard deviations; for a review of the concept of statistical power, see our companion paper [16] or [30].

For simplicity, we compute statistical power for simple hypotheses only (*e.g.* BCa English versus uniform noise), where both languages are known. Let ψ denote any test statistic, and suppose we wish to apply ψ to distinguish any two languages L_0 and L_1 at a given string length. That is, given any string Y of the specified length, we wish to test the null hypothesis H_0 that Y was produced by L_0 versus the alternative hypothesis H_1 that Y was produced by L_1. Let $\mu_0 \pm \sigma_0$ and $\mu_1 \pm \sigma_1$ be the sample mean and standard deviation of ψ when applied, respectively, to strings from L_0 and L_1 of the given length, and assume that $\mu_0 < \mu_1$. We compute power from the values $\mu_0, \sigma_0, \mu_1, \sigma_1$; thus, in our power calculations, we equate H_0 with the parameters μ_0, σ_0, and H_1 with μ_1, σ_1.

Let $0 < \alpha < 1$ be any critical level. The power of ψ at critical level α is one minus the type II error (accepting H_0 when H_1 is true) rate at the specified critical level. Using a one-sided test, we reject H_0 whenever $\psi(Y) > \tau_\alpha$, where the threshold τ_α is chosen such that the type I error (rejecting H_0 when H_0 is true) rate $\text{Prob}[\psi > \tau_\alpha | H_0] \leq \alpha$. Thus, for the simple hypothesis H_1, the *power of ψ at critical level α* is $\beta = 1 - \text{Prob}[\psi < \tau_\alpha | H_1]$. Equivalently, power is the area under the probability density function for $\psi | H_1$ to the right of the threshold τ_α. For any given critical level, it is desirable to maximize power.

We approximate β from α and the sample means and standard deviations $\mu_0 < \mu_1$, $\sigma_0 > 0$, $\sigma_1 > 0$ of ψ as follows. To begin, let Z denote the standard normal distribution. Also, for any real number x, let $\Phi(x) = \text{Prob}[Z > x] = (1/\sqrt{2\pi}) \int_x^{+\infty} e^{-x^2/2} dx$ be the area of the right tail of the standard normal distribution, and let Φ^{-1} be the inverse of Φ. First, assuming $(\psi - \mu_0)/\sigma_0 \approx Z$ under H_0, we compute the threshold $\tau_\alpha = \Phi^{-1}(\alpha)\sigma_0 + \mu_0$ so that $\text{Prob}[\psi > \tau_\alpha | H_0] \approx \alpha$. Second, assuming $(\psi - \mu_1)/\sigma_1 \approx Z$ under H_1, we compute $\beta = 1 - \text{Prob}[\psi < \tau_\alpha | H_1] = \text{Prob}[\psi > \tau_\alpha | H_1] \approx \text{Prob}[Z > (\tau_\alpha - \mu_1)/\sigma_1] = \Phi(\Phi^{-1}(\alpha)\sigma_0/\sigma_1 - (\mu_1 - \mu_0)/\sigma_1)$. A straightforward variation of this procedure is used when $\mu_0 > \mu_1$.

Note that power is not necessarily symmetrical with respect to H_0 and H_1. Thus, the power of ψ at distinguishing BCa English from uniform noise is not necessarily equal to the power of ψ at distinguishing uniform noise from BCa English. This asymmetry can result from differences in the dispersions between ψ given H_0 and ψ given H_1.

Our method for computing power involves an approximation: we base our calculations on the observed first two moments of ψ only, computed from 10,000 randomly chosen strings. More specif-

[9] The *observed power* of a test statistic is its power observed from our experimental data, as opposed to its theoretical power.

ically, we assume $(\psi - \mu_0)/\sigma_0 \approx Z$ under H_0 and $(\psi - \mu_1)/\sigma_1 \approx Z$ under H_1, the appropriateness of which assumptions can be assessed from the histograms. Since we calculate statistical power separately for each string length, however, our calculations do not depend on any other assumption, such as the asymptotic χ^2 distribution of X^2. Although we could have calculated observed power exactly from the raw data that produced the histograms, we chose not to for simplicity and since we believe our method yields an approximation good enough for most engineering decisions.

We calculated power at the four critical levels 0.1, 0.01, 0.001, 0.0001, computing tail areas of Gaussian distributions with the erf function from our C math library and with the SAS statistical package [34]. In particular, we used $\Phi^{-1}(0.1) = 1.28155$, $\Phi^{-1}(0.01) = 2.32635$, $\Phi^{-1}(0.001) = 3.09023$, and $\Phi^{-1}(0.0001) = 3.71902$.

3.6 Our Language-Recognition Toolkit

To carry out our experiments, we designed and implemented a toolkit of C programs running under the DEC Ultrix and SGI Irix operating systems. These programs include routines to remove extraneous characters from text files, to count k-gram frequencies, to simulate any Markov chain, and to display data. Particular experiments are run by a program that selects strings from a text file at random and applies tests to these strings. All experiments were carried out on 32-bit machines using double-precision arithmetic operators of the C programming language. To select sample strings at random from each candidate language, to generate random noise, and to drive our our Markov chain simulators, we used the pseudorandom number generator "random()".

4 Results

We now describe and compare the distributions of our test statistics when applied to strings of various lengths drawn from the nine test languages. Specifically, we summarize and explain the results of Experiments 1 and 2 and of our power calculations. Finally, we discuss some issues raised in these findings. Throughout, the base language is BCa.

Complete experimental results are given in Appendix B and in the six supplemental appendices [17]. In particular, Appendix B lists detailed results for Experiment 1 on plaintext length: for each test statistic, there is a table of sample means and standard deviations of the test statistic applied to each test language for various string lengths. The amount of data is overwhelming, and it is difficult to intuit phenomena from these numerical tables. Therefore, in this section, we selectively present highlights of our data through several graphs, histograms, and tables of statistical power. Additional similar figures and tables are included in supplemental Appendices D, E, and F.

4.1 Results of Experiment 1 (plaintext length)

We describe the distributions of the test statistics in histograms and in graphs of mean $\pm$ standard deviation. The histograms provide thorough descriptions of the distributions at fixed string lengths; the graphs summarize this information showing the effect of string length.

Histograms

Figures 1–6 describe the distributions of the statistics S/N, IC, X^2, ML, and IND when applied to BCa English, 1st-order BCa English, 0th-order BCa, and uniform noise. For example, for each of these four test languages, Figure 1 gives histograms of 100 observed values of S/N computed from randomly selected strings of length $\lg n = 10$, and Figure 2 gives histograms of S/N for $\lg n = 4$. Figures 3–6 present similar histograms of IC, X^2, ML, and IND for $\lg n = 10$. These histograms reveal the shape, dispersion, and separation of the distributions.

As explained in Section 3.1, BCa is a type of real English; 1st-order BCa is an idealized model of BCa; and 0th-order BCa and uniform noise are alternatives to English. When distinguishing BCa English from uniform noise, ideally we would like the distribution of the statistic on BCa to be perfectly separated from its distribution on uniform noise. If a statistic is robust (with respect to our 1st-order model), its distributions for BCa and 1st-order BCa will be nearly identical.

As shown by the separations of the distributions, the relative effectiveness of test statistics to distinguish pairs of languages depends crucially on the languages. For example, at $\lg n = 10$, all statistics perfectly separated BCa English from uniform noise (and 1st-order BCa English from 0th-order BCa). But IC, ML, and IND more cleanly separated 0th-order BCa from uniform noise than did S/N and X^2—in particular, the distributions of S/N on 0th-order BCa and uniform noise overlapped substantially.

The histograms of S/N with $\lg n = 4$ in Figure 2 illustrate a general trend that, for smaller string sizes, the distributions overlap more and become more disperse. Our power calculations quantify the effect of this overlap on the ability of the test statistics to distinguish pairs of languages.

The histograms graphically show significant differences in the dispersions of the distributions. For example, the IC has a very narrow dispersion for uniform noise, and a much greater dispersion for BCa English. By contrast, at $\lg n = 10$, S/N, X^2, and ML had greater dispersions for uniform noise than for BCa English. This behavior is consistent with the intended purpose of the statistics and confirms that IC is relatively more appropriate for recognizing uniform noise than for recognizing BCa English, and that S/N, X^2, and ML are relatively more appropriate for recognizing BCa English than for recognizing uniform noise. Thus, when using the IC at any critical level, the threshold can be set absolutely closer to the mean of IC given H_0 when distinguishing uniform noise from BCa English than when distinguishing BCa English from uniform noise, resulting in higher power. Since many cipher systems yield uniform text when decrypted with incorrect keys, the IC is well-suited for detecting unknown nonuniform plaintext.

Interpreting the IC as a measure of the roughness of a distribution yields an intuitive explanation of the IC histograms on BCa English and uniform noise. The higher mean value of IC on BCa English indicates that the bigram distribution for BCa is rougher than that for uniform noise. The small variance of IC on uniform noise indicates that the variation in this roughness over multiple samples is small on uniform noise.

As we would like, the histograms for 1st-order BCa are very similar to those for BCa. Thus, with respect to our bigram statistics, our simulated 1st-order BCa English closely models real BCa English. In this sense, the statistics are robust. This robustness observation, however, does not imply that our 1st-order model of English is an accurate model of English when higher-order features can be extracted from the strings. Although BCa and 1st-order BCa yield similar distributions, there are notable differences: the distributions for 1st-order BCa are slightly more symmetrical and more bell-shaped than are those for BCa. Similarly, the distributions for 0th-order BCa tend to be slightly more symmetrical and bell-shaped than are those for uniform noise.

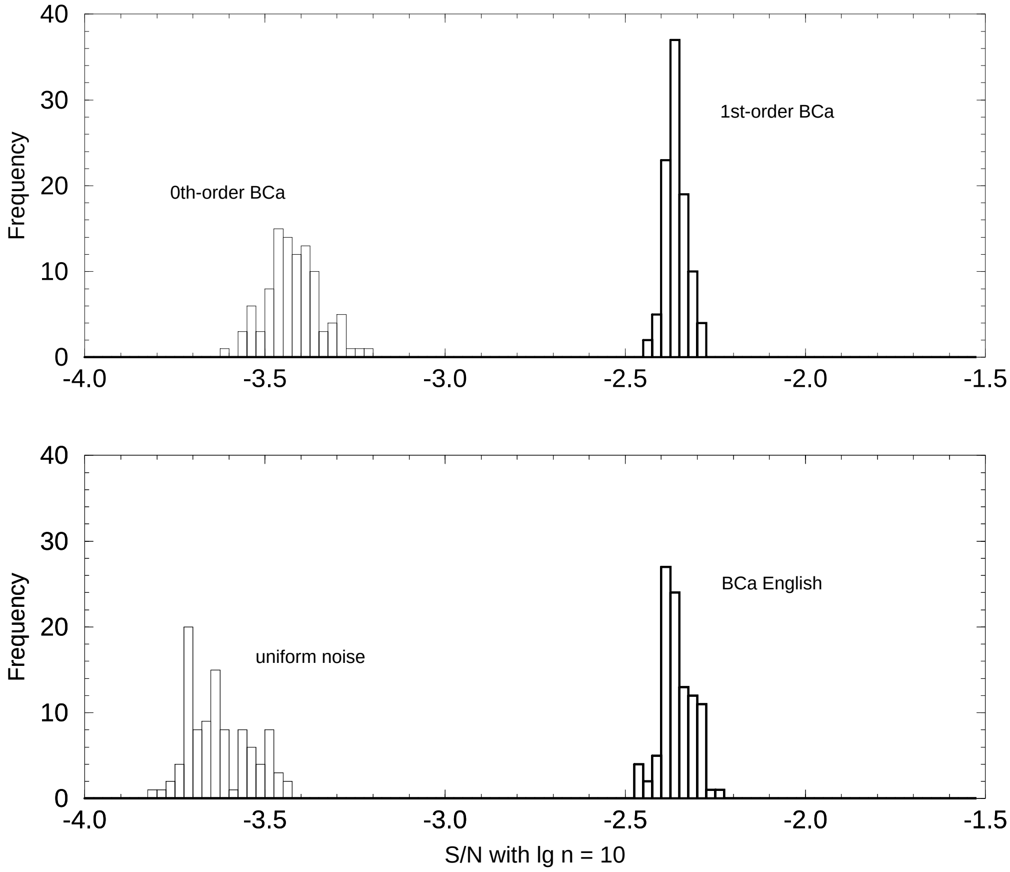

Figure 1: Four histograms of values of the statistic S/N computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English, 1st-order BCa English, 0th-order BCa noise, and uniform noise, respectively. For example, for uniform noise, 20 of the chosen strings had S/N values between -3.700 and -3.725. Base language is BCa.

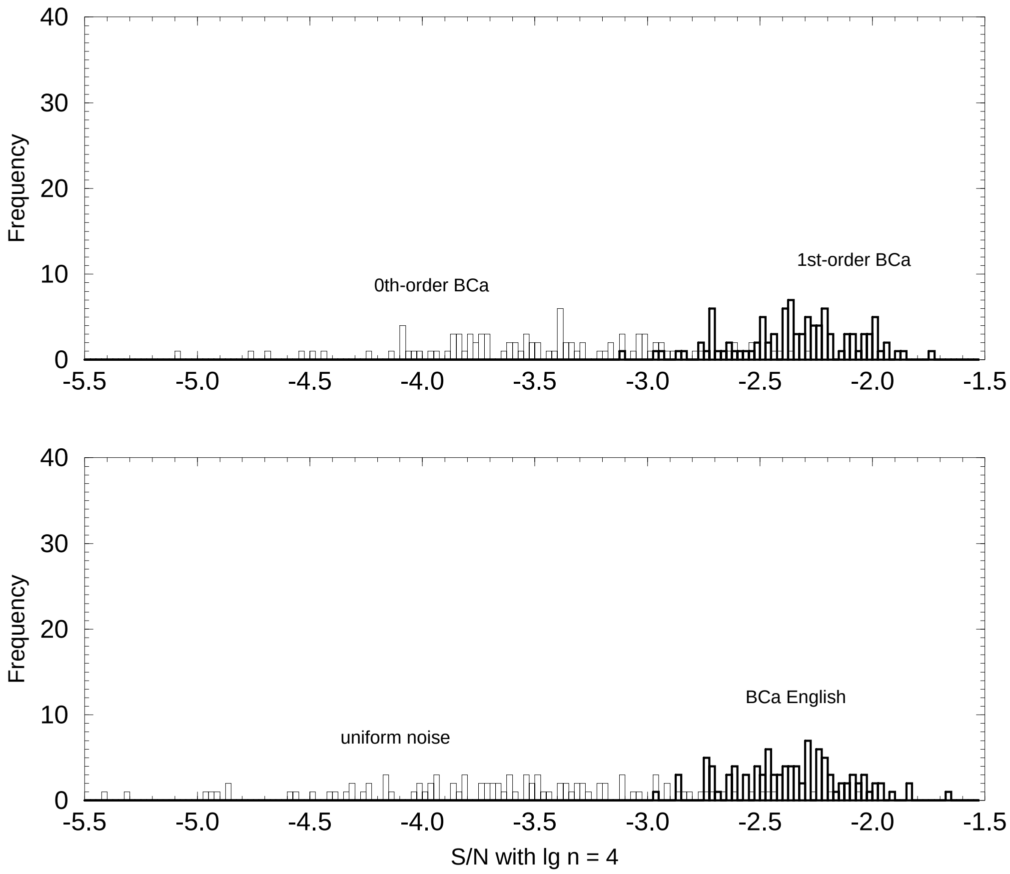

Figure 2: Four histograms of values of the statistic S/N computed on 100 randomly chosen strings of length $\lg n = 4$ drawn from BCa English, 1st-order BCa English, 0th-order BCa noise, and uniform noise, respectively. Base language is BCa.

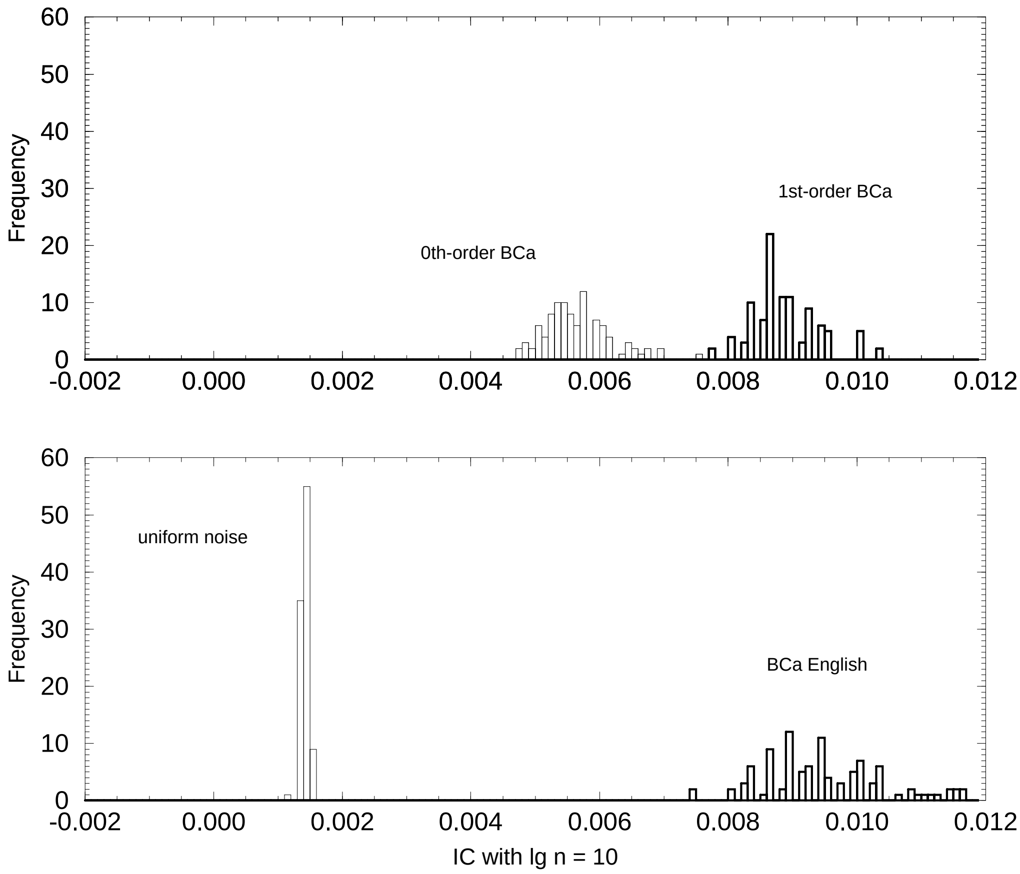

Figure 3: Four histograms of values of the statistic IC computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English, 1st-order BCa English, 0th-order BCa noise, and uniform noise, respectively.

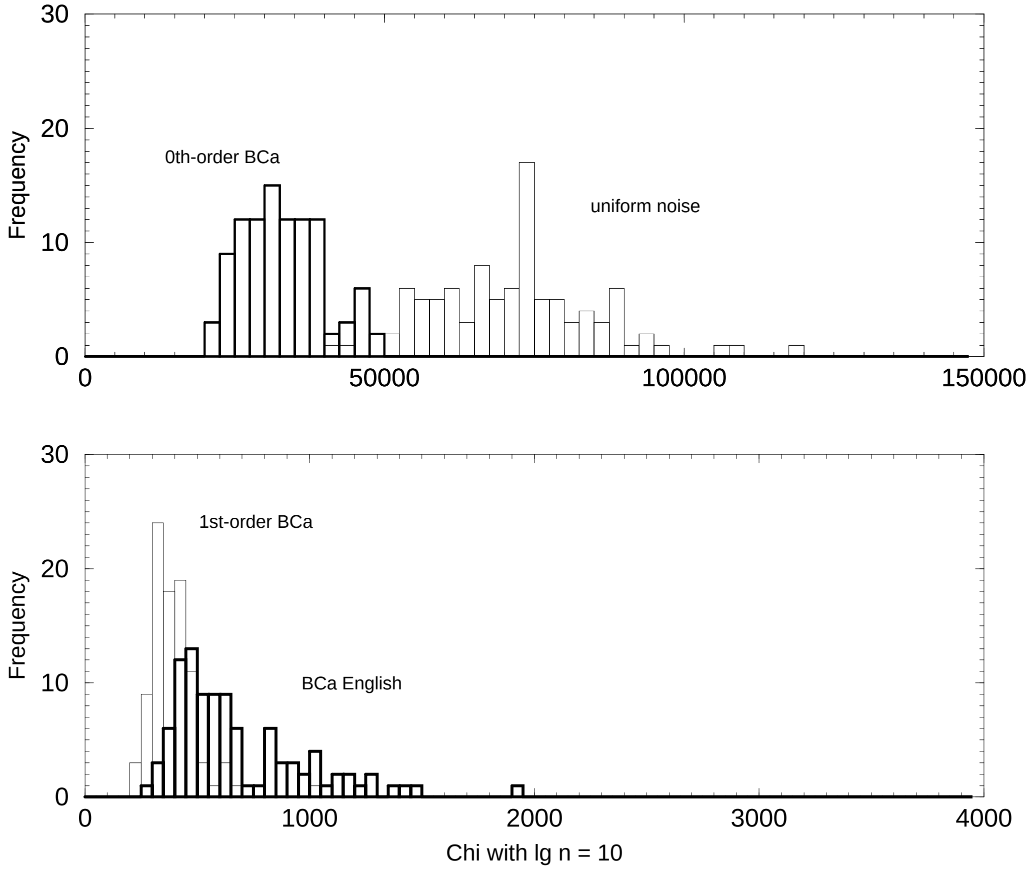

Figure 4: Four histograms of values of the unnormalized statistic X^2 computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English, 1st-order BCa English, 0th-order BCa noise, and uniform noise, respectively. Due to the extreme range of values for this unnormalized statistic, the horizontal axis of each panel is drawn with a different scale. Indeed, X^2 very effectively separates BCa English from uniform noise. Base language is BCa.

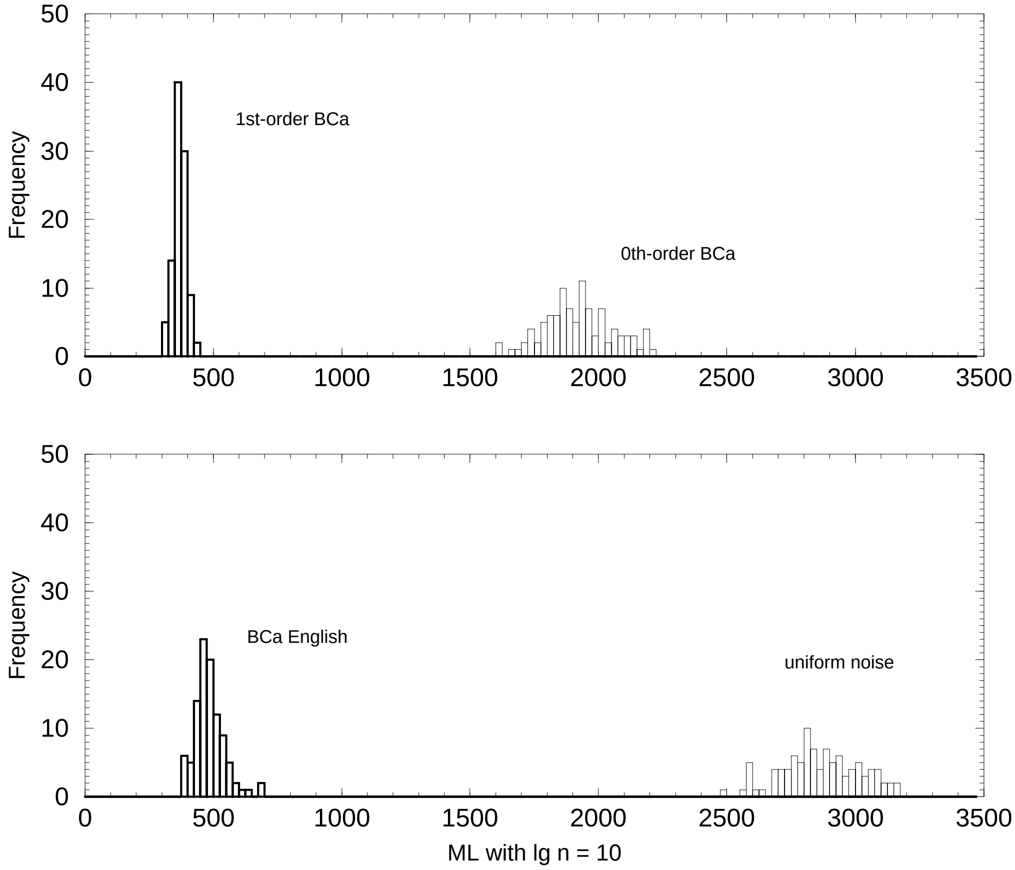

Figure 5: Four histograms of values of the unnormalized statistic ML computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English, 1st-order BCa English, 0th-order BCa noise, and uniform noise, respectively. Base language is BCa.

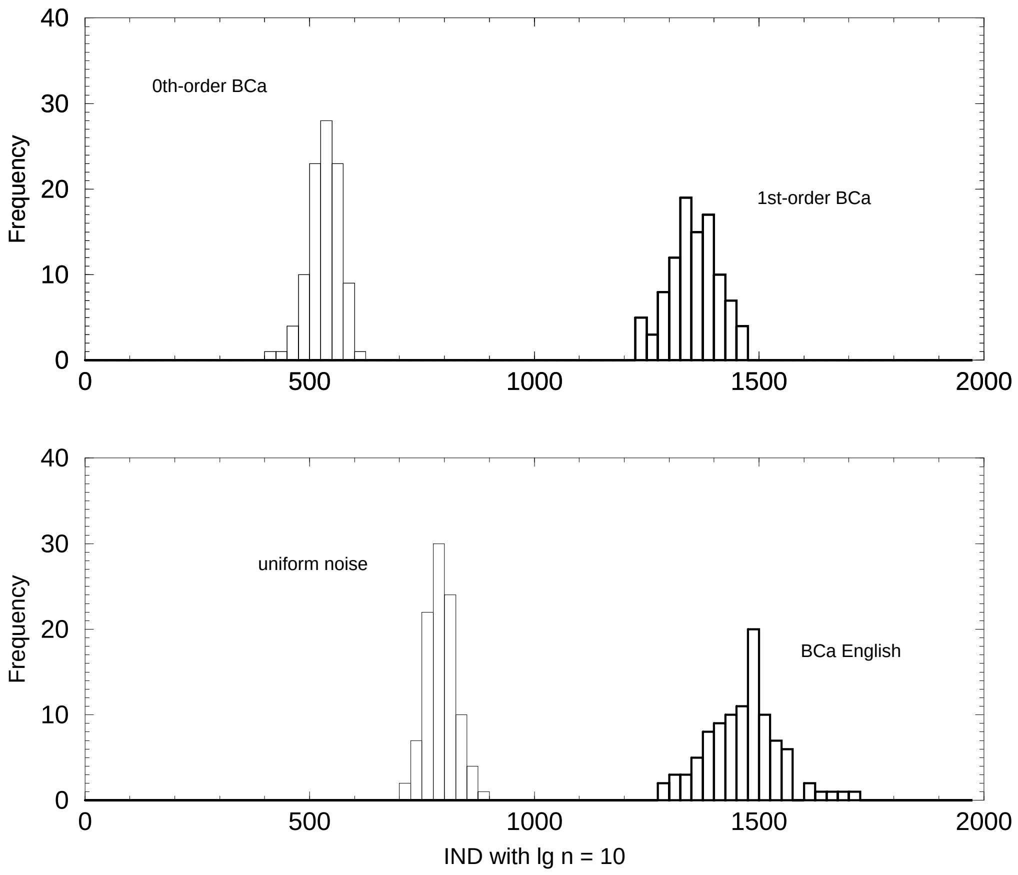

Figure 6: Four histograms of values of the unnormalized statistic *IND* computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English, 1st-order BCa English, 0th-order BCa noise, and uniform noise, respectively.

Means and Standard Deviations

To examine the effects of string length on the test statistics, it is convenient to summarize the distributions in terms of their first two moments. We now do so, focusing on the statistics $\hat{S}$, IC, and $\diamond X^2$. Figures 7, 8, and 9 show, respectively, how the means of $\hat{S}$, IC, and $\diamond X^2$ vary with string length for all nine test languages. Figures 10 and 11 show the observed means and standard deviations for $\hat{S}$ and $\hat{IC}_\star$ when applied to BCa English and uniform noise; these standard deviations are crucial in comparing the effectiveness of the test statistics.

Figure 7 illustrates that, for sufficiently large strings, the mean of $\hat{S}$ differs for most of the test languages. For example, the three alternatives to English (uniform noise, 0th-order BCa, and "er" repeated) rapidly diverge from the other test languages. The significance of these differences is established by the standard deviations, as depicted in Figure 10 and given in Appendix B. By $\lg n = 8$, the mean of $\hat{S}$ separates the remaining test languages into three groups: real and simulated BCa English, WSJ1, and the remaining three types of real and simulated English. Figure 8 shows a similar trend for IC. The IC, however, had a more chaotic behavior for very short strings—perhaps because very short strings are unlikely to have repeated bigrams. Also, the IC more cleanly separated uniform noise from 0th-order BCa than did $\hat{S}$, which is useful to know for cryptanalytic applications that yield unigrams only.

Figures 10 and 11 illustrate the behavior of $\hat{S}$ and $\hat{IC}_\star$ at distinguishing BCa English from uniform noise at various string lengths. Although both statistics eventually perfectly separate these two languages, $\hat{S}$ was relatively more effective at small lengths, as is quantified in our power calculations (see Figure 14). In addition, as observed in Figure 3, Figure 11 shows that $\hat{IC}_\star$ has a smaller standard deviation on uniform noise than on BCa English.

Our normalizations $\diamond X^2$ and $\hat{S}$ of X^2 and S, respectively, were intended to produce statistics whose distributions are approximately standard normal when applied to 1st-order BCa (and BCa). Similarly, $\hat{IC}_\star$ was intended to have an approximately standard normal distribution when applied to uniform noise (see Section 2). Figures 9–11 show that this intent was achieved fairly well for $\hat{IC}_\star$ and $\hat{S}$ but not for $\diamond X^2$, except when $\diamond X^2$ was applied to long strings of 1st-order BCa. Nevertheless, as proven in the power tables, the relative separations of the means of $\diamond X^2$ on most languages are indeed significant. Thus, when interpreting the values of $\diamond X^2$, it is important to use experimentally-determined thresholds. Similar statements also apply to $\diamond ML$ and $\diamond IND$.

As shown in Figure 10 and Appendix B, although $\hat{S}$ is approximately standard normal on BCa, its standard deviation increases slightly with string length throughout. We believe this slight increase in the standard deviation of $\hat{S}$ results from $\hat{S}$ being applied to dependent bigrams and from the fact that BCa English is not a 1st-order Markov chain. By contrast, when applied to 1st-order BCa, the standard deviation of $\hat{S}$ remains approximately constant and slightly less than 1. For huge strings, however, the standard deviation of $\hat{S}$ on 1st-order BCa decreases slightly, we believe reflecting the bias of selecting the sample strings from the base sample.

Table 8 of supplemental Appendix B reveals some interesting properties of the $\diamond X^2$ statistic. First, from these means and standard deviations, it is apparent that a standard normal interpretation of $\diamond X^2$ is not useful, except for recognizing long strings of 1st-order BCa. For example, the observed means of $\diamond X^2$ on BCa lie in the interval $[-5, 5]$ only for string lengths $7 \leq \lg n \leq 10$, even though BCa is the base language and the candidate strings were drawn from the same base sample used to compute the BCa transition probabilities. Even for 1st-order BCa, the values of $\diamond X^2$ fall outside of $[-5, 5]$ until $\lg n \geq 7$. Second, for long strings, the values of $\diamond X^2$ diverge for all languages except 1st-order BCa. For example, at $\lg n = 17$, the mean value of $\diamond X^2$ is approximately 25 for BCa, 80 for BCg, and 3,312 for uniform noise. Although $\diamond X^2$ grows more slowly on BCa than on BCg, it is interesting that $\diamond X^2$ nevertheless diverges even when the candidate and base languages are the same. Similar trends also occur for $\diamond ML$.

The first observation might be explained in part by inaccuracies in our normalization of X^2 for short strings. In addition, to interpret X^2 with the χ^2 distribution, some statisticians recommend that the expected number of observations in each cell (bigram) be at least 5. With 534 degrees of freedom, and assuming a uniform distribution of bigrams, it would take a string of length $534 \cdot 5 = 2,670 > 2^{11}$ to satisfy this rule. Nevertheless, our results demonstrate the well-known fact that X^2 can be effectively used at a variety of string lengths, even when the asymptotic χ^2-distribution does not hold. For example, our power calculations show that $\diamond X^2$ is extremely effective at distinguishing BCa English from uniform noise when computationally-determined thresholds are used.

The second observation reflects the folklore that X^2 always rejects for large n because the model is never fully satisfied. In particular, only 1st-order BCa (and not BCa), fully satisfies the model. Due to this sensitivity of the X^2 statistic, practitioners should exercise care when recognizing a known language.

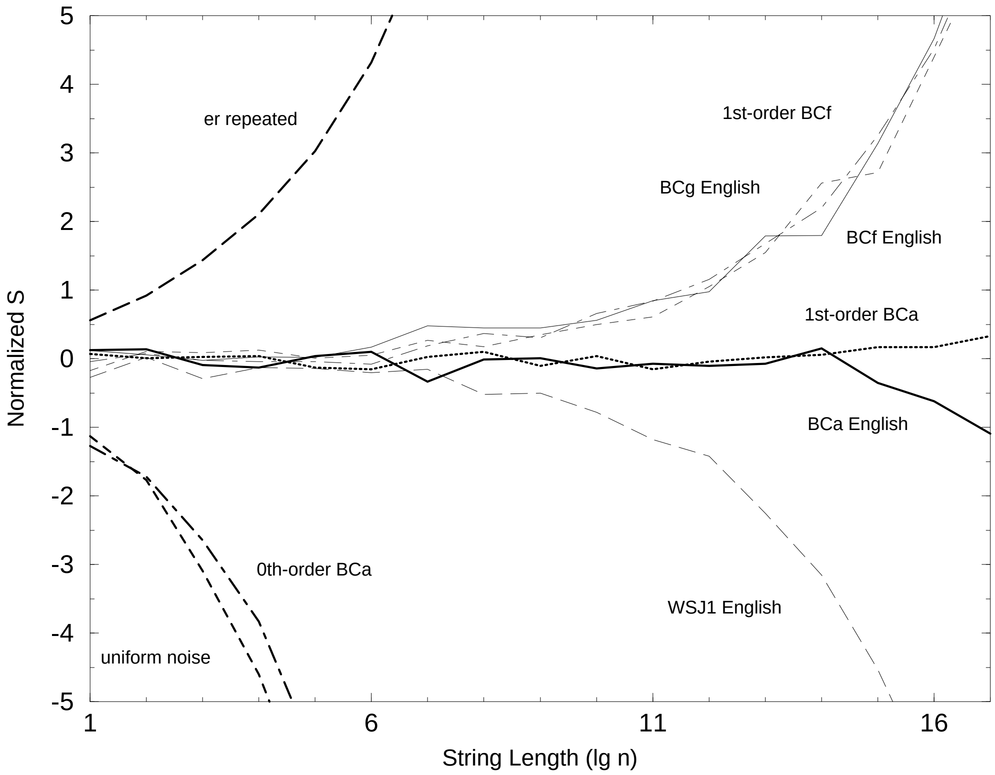

Figure 7: Sample means of the statistic $\hat{S}$ computed on 100 randomly chosen strings of various lengths drawn from each of the nine test languages, respectively. Base language is BCa.

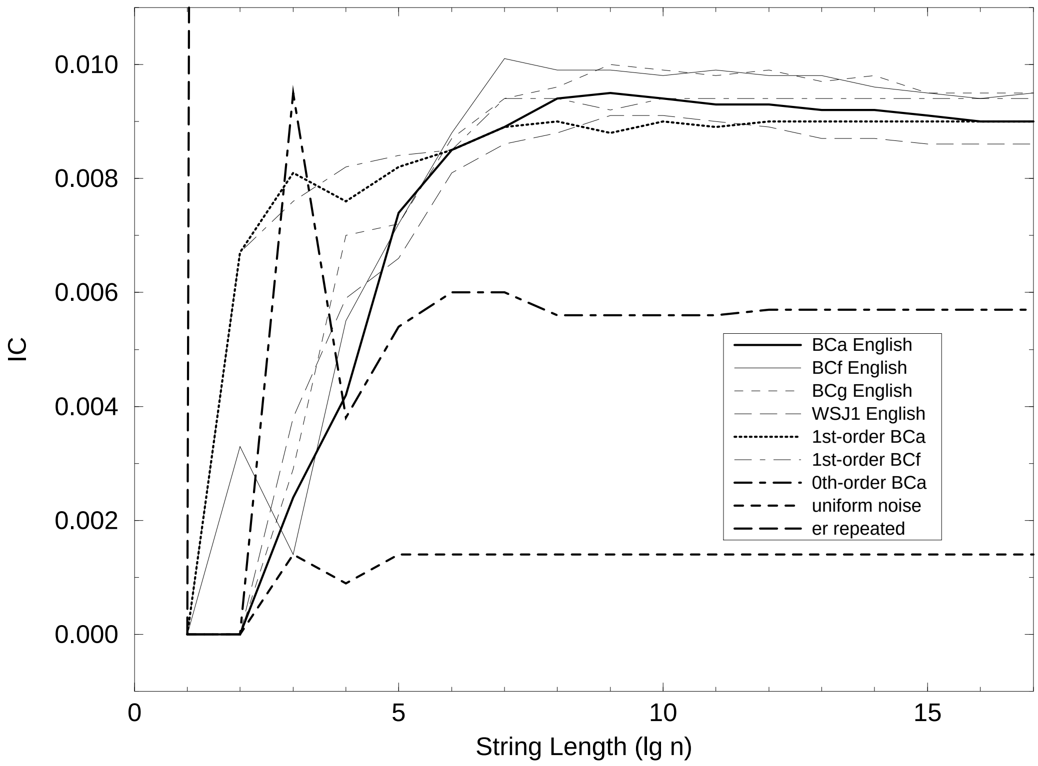

Figure 8: Sample means of the statistic IC computed on 100 randomly chosen strings of various lengths drawn from each of the nine test languages, respectively.

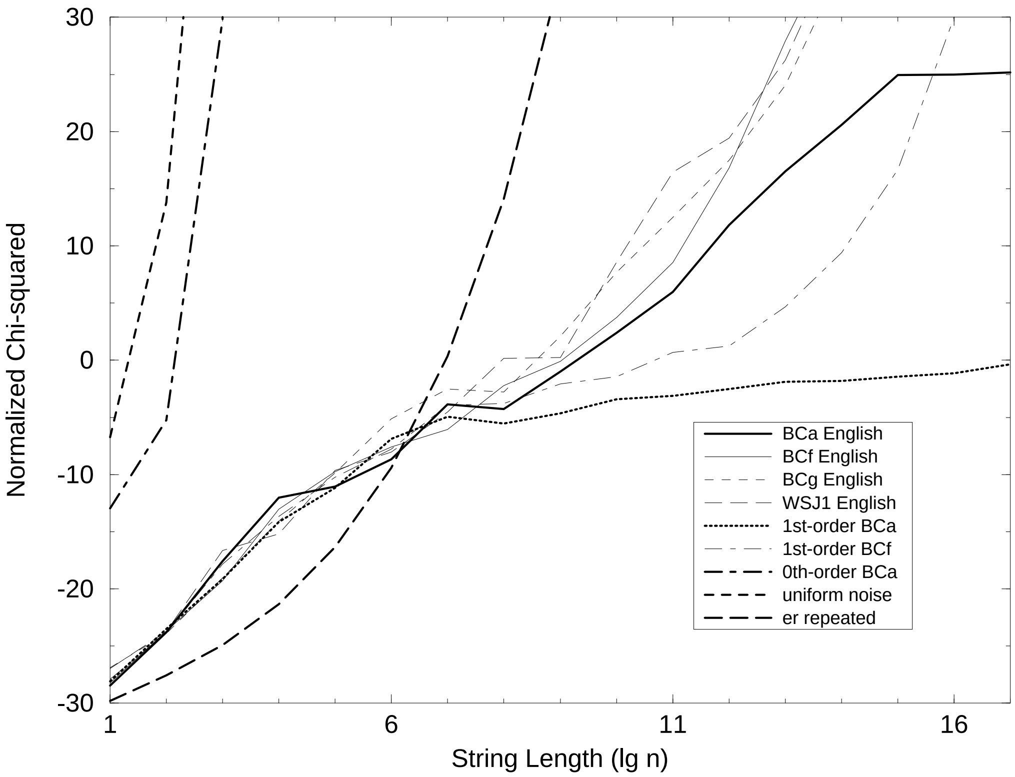

Figure 9: Sample means of the statistic $\diamond X^2$ computed on 100 randomly chosen strings of various lengths drawn from each of the nine test languages, respectively. Base language is BCa.

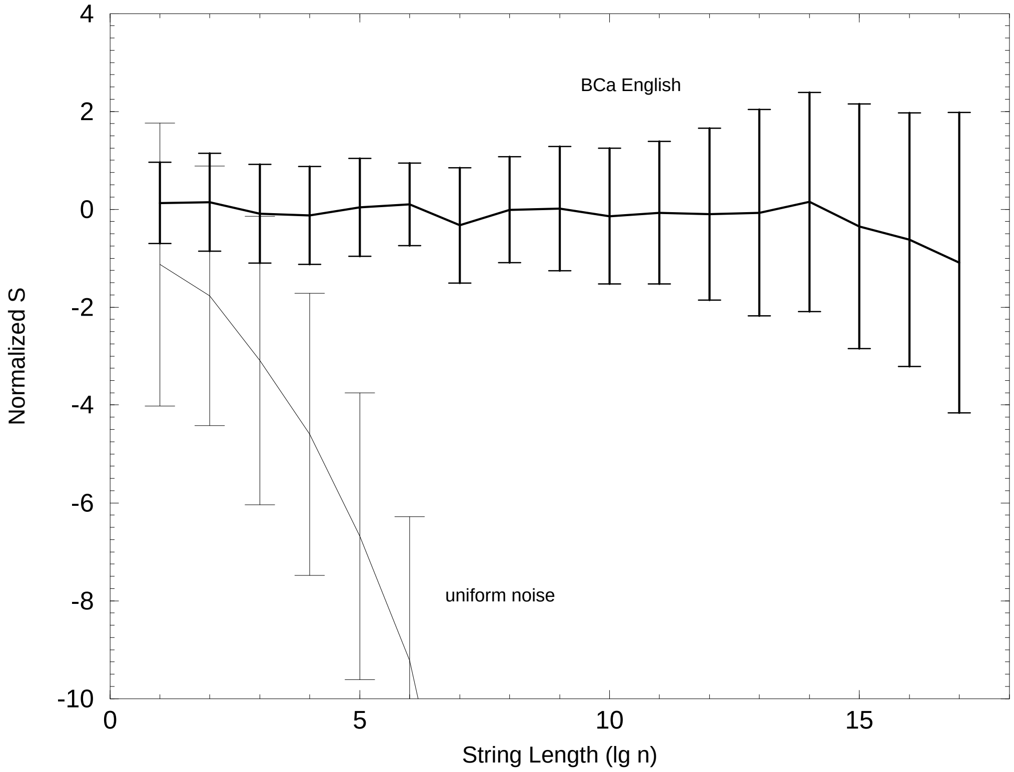

Figure 10: Sample means and standard deviations of the statistic $\hat{S}$ computed on 100 randomly chosen strings of various lengths drawn from BCa English and uniform noise, respectively. Base language is BCa.

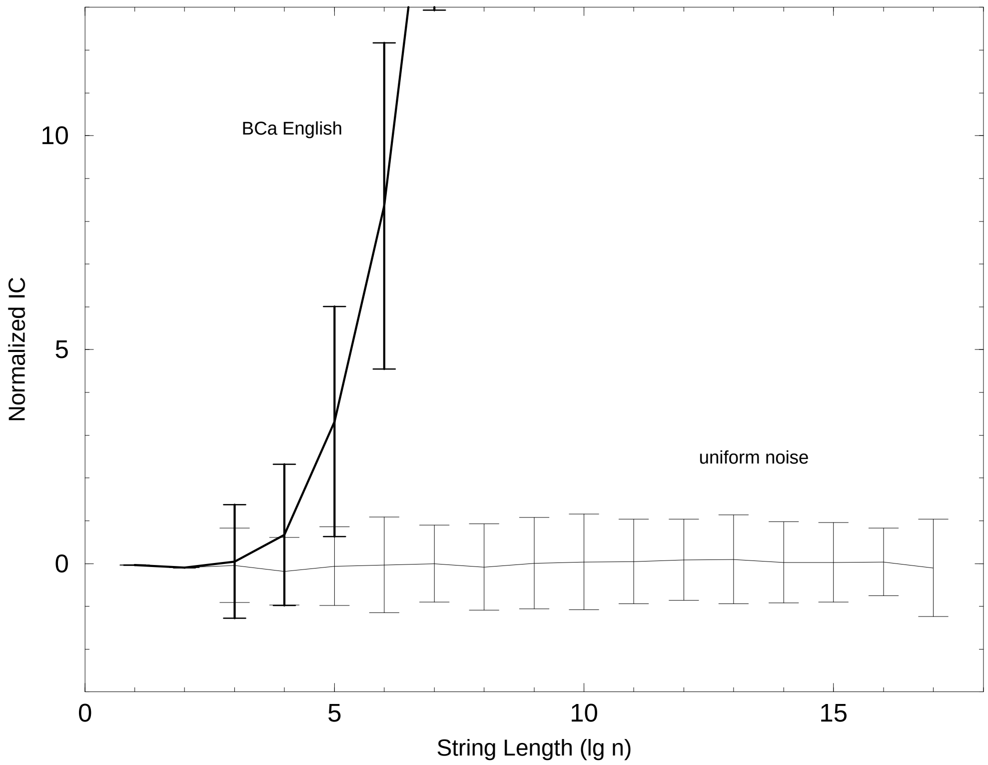

Figure 11: Sample means and standard deviations of the statistic $\hat{IC}_\star$ computed on 100 randomly chosen strings of various lengths drawn from BCa English and uniform noise, respectively.

4.2 Results of Experiment 2 (noisy plaintext)

Figures 12 and 13 highlight the results of Experiment 2 by illustrating the behavior of $\hat{S}$ and $\hat{IC}_\star$ at distinguishing BCa English from uniform noise in the presence of noise. Specifically, these figures show the first two moments of $\hat{S}$ and $\hat{IC}_\star$ when applied to strings of length $\lg n = 10$ drawn from BCa English and uniform noise with various amounts of added uniform noise. In these figures, the curve for uniform noise serves as a reference line, which could have been drawn using data from Experiment 1 or from theoretically computed values.

At $\lg n = 10$, $\hat{IC}_\star$ is able to distinguish BCa English from uniform noise in the presence of higher amounts of noise than is $\hat{S}$. For example, the standard deviation bars for $\hat{S}$ in Figure 12 first overlap at 65% added noise, whereas those for $\hat{IC}_\star$ first overlap at 80% added noise in Figure 13. This behavior is quantified in our power calculations (see Figure 18).

Figures 12 and 13 also show another difference between the statistics $\hat{S}$ and $\hat{IC}_\star$: with increasing noise levels, $\hat{IC}_\star$ on noisy BCa converges at a faster rate to $\hat{IC}_\star$ on uniform noise than does $\hat{S}$, perhaps reflecting the quadratic nature of the IC formula.

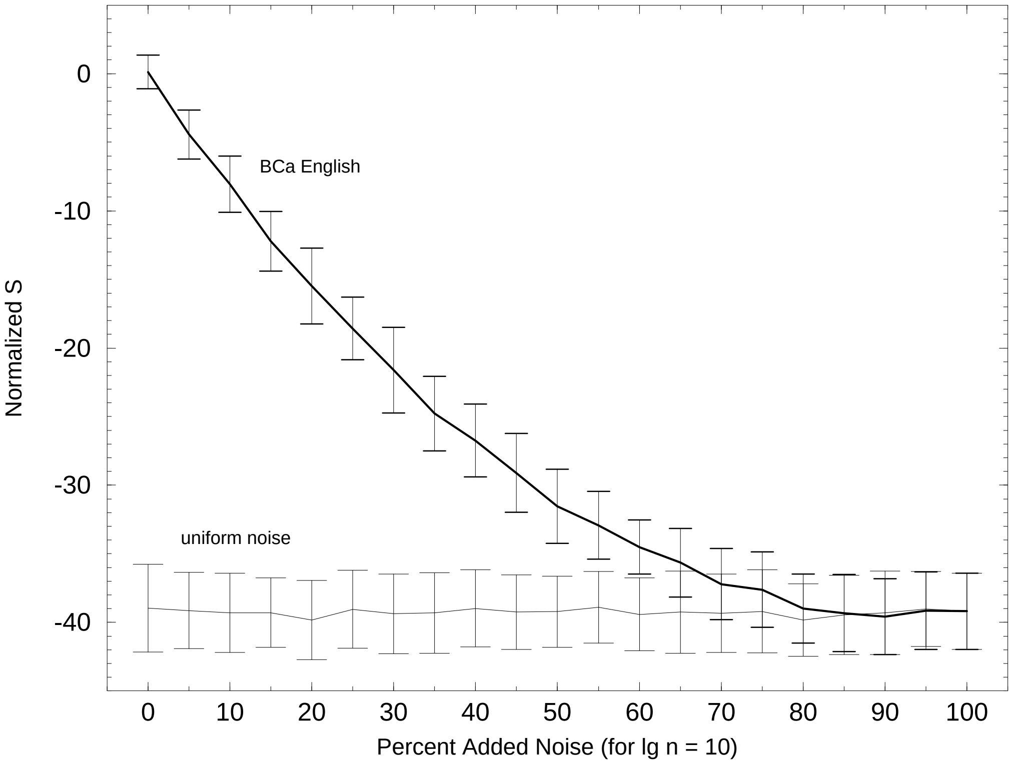

Figure 12: Sample means and standard deviations of the statistic $\hat{S}$ computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English and uniform noise, respectively, with various amounts of added uniform noise. Base language is BCa.

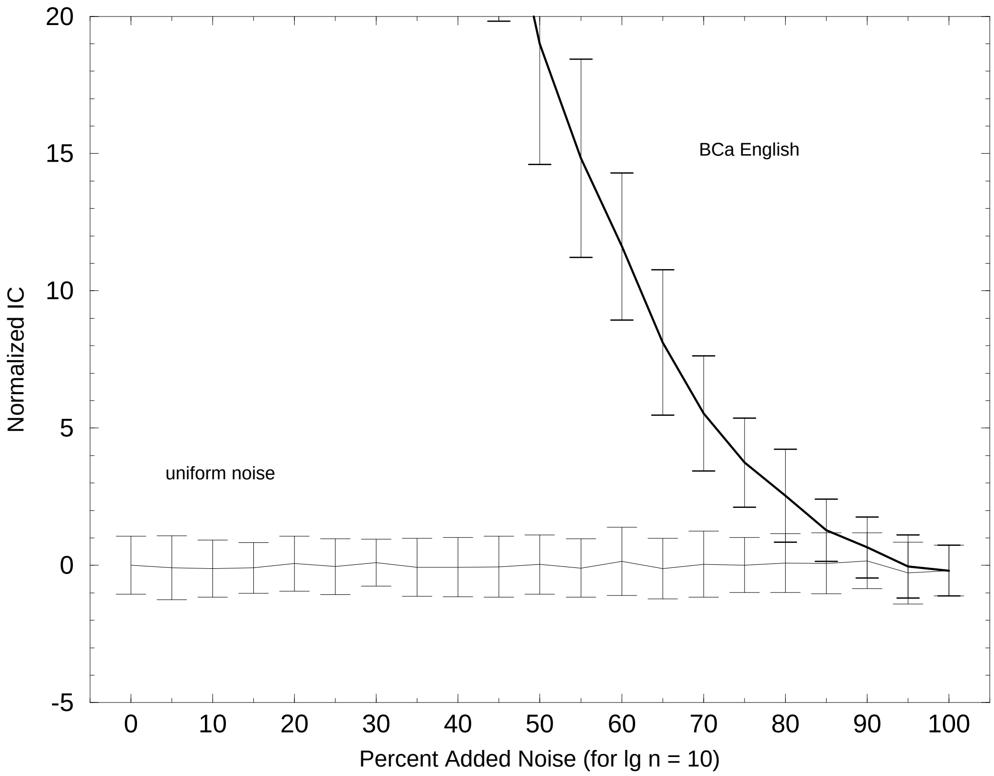

Figure 13: Sample means and standard deviations of the statistic $\hat{IC}_\star$ computed on 100 randomly chosen strings of length $\lg n = 10$ drawn from BCa English and uniform noise, respectively, with various amounts of added uniform noise.