Probabilistic Models of Grammar:
Phonology as Information Minimization

John Goldsmith
The University of Chicago

Keywords: probabilistic phonology Japanese English entropy

Two philosophers who disagree about a point should, instead of arguing fruitlessly and endlessly, be able to take out their pencils, sit down amicably at their desks, and say "Let us calculate."
Gottfried Wilhelm von Leibniz (1646 – 1716)

Lest men suspect your tale untrue,
Keep probability in view.
John Gay (1685–1732)
Fables. Part 1: The Painter who pleased Nobody and Everybody.

But to us, probability is the very guide of life.
--Bishop Joseph Butler, *The Analogy of Religion*, Introduction

It is seen in this essay that the theory of probabilities is at bottom only common
sense reduced to calculus; it makes us appreciate with exactitude that which exact
minds feel by a sort of instinct without being able ofttimes to give a reason for it.
Marquis Pierre-Simon de Laplace
Philosophical Essay on Probabilities (1814)

In the description that follows, language will be treated as a Markoff process. The phonemes will be considered uniquely identifiable; but their order, in the sequences that compose our sample, can be described only statistically.
Colin Cherry, Morris Halle, and Roman Jakobson
Toward the logical description of languages in their phonemic aspect (1953)

Water which is too pure has no fish.
Ts'ai Ken T'an

1. Introduction

My goal in this paper is to provide an introduction to the notion of a probabilistic grammar, and in particular, a probabilistic phonology.[1] The notion of a probabilistic grammar is not a new one; it originated in the 1950s in work by Ray Solomonoff and others, and has played an increasingly important role in computational syntax and in speech recognition over the last fifteen years (Solomonoff 1997,[2] Charniak 1993). The notion of a probabilistic grammar is, however, relatively unknown in mainstream linguistics – both syntax and phonology – and this is

This paper appeared in *Phonological Studies* #5: 21-46 (2002).

unfortunate, for I believe that the ideas involved here are extremely fruitful for understanding various problems in linguistics.

In this paper, I will focus on phonology from a probabilistic point of view. Both probabilistic phonology and morphology (Goldsmith 2001b) are areas with relatively little work done to date. In fact, the only work that I am aware of in probabilistic phonology in the last thirty years or so is Coleman and Pierrehumbert 1997, other than the voluminous literature on speech recognition using hidden Markov models and the like.[3]

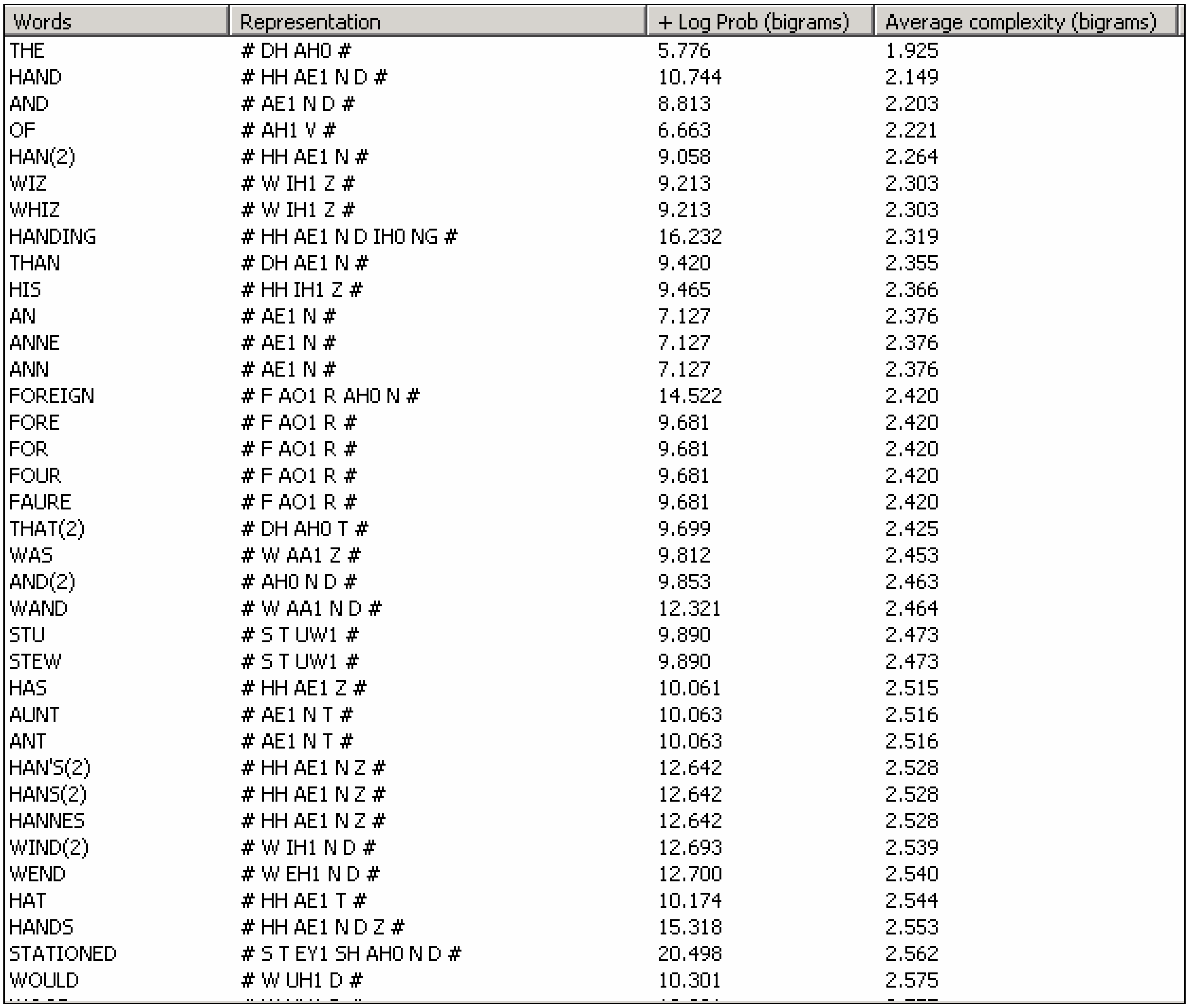
To illustrate the basic ideas, I have assembled a simple linear-segmental model of phonology and applied notions of probability theory to it, and put it all into a computer program which we will have occasion to look at; some of its output is given in this paper, and it can be downloaded from http://humanities.uchicago.edu/faculty/goldsmith/Chiba.

This program (which I will call a "Complexity Sorter") takes as its input a list of words from a given language, with both standard orthography and phonological representation, and also word frequency if that is available. It accepts this material from a computer file, and produces various graphical outputs. In particular, it calculates what I will call the "phonological complexity" of each word – essentially the average *information* content, from the point of view of information theory – and sorts the words by this measure (following standard usage, *information* and *complexity* may be in some contexts used interchangeably). We may then browse through the words of the language from the top of the list to the bottom. The program performs the analysis with no prior knowledge of the language.

These lists are illustrated in Tables 1 through 4 for English and Japanese. Table 1 is the "good" end of the English list, the words which have the lowest "average complexity" in these tables, a notion closely tied to probability and one which we will discuss in detail shortly. The "good" end, then, has low average complexity, and consists of words whose phonology is completely native and central to the phonology of the language: words like *the, hand, of, and*, etc.

The notation used here may not be familiar. The word in normal orthographic form is given in the first column, and a phonological representation is given in the second column. The notation that is used here is the one that is commonly used in the computational literature for English phonemes. It has the advantage over the IPA system that is more familiar to us that it only uses the basic 26 letters of the alphabet, but many of the symbols used for the phonemes are less obvious. For example, the voiced fricative of "the" is spelled "DH", while the schwa is spelled "AH0"; more generally, every vowel ends with a number which indicates its stress level. Vowels begin with two letters; if the second letter is H, the vowel is lax (as in IH, EH, and so on), while if the second letter is Y or W, the letter is a tense diphthong), etc. This is widely known as the "DARPAbet" notation. (see, for example, Jurafsky and Martin 2000.).

The bottom (or "bad" end) of the English wordlist – the words with the lowest average probability, indicated here as the highest "average complexity" is given in Table 2.

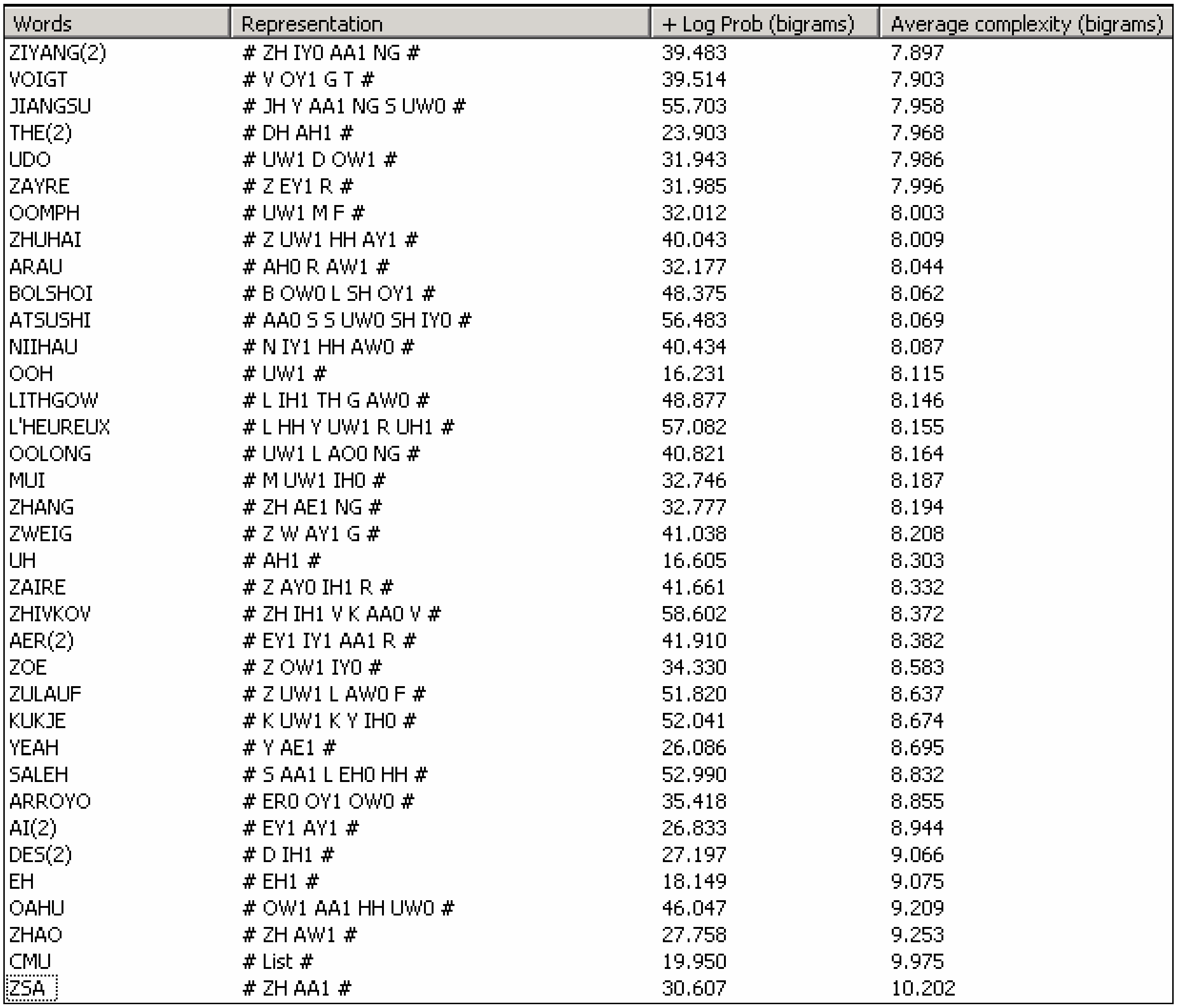


Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)
THE	# DH AH0 #	5.776	1.925
HAND	# HH AE1 N D #	10.744	2.149
AND	# AE1 N D #	8.813	2.203
OF	# AH1 V #	6.663	2.221
HAN(2)	# HH AE1 N #	9.058	2.264
WIZ	# W IH1 Z #	9.213	2.303
WHIZ	# W IH1 Z #	9.213	2.303
HANDING	# HH AE1 N D IH0 NG #	16.232	2.319
THAN	# DH AE1 N #	9.420	2.355
HIS	# HH IH1 Z #	9.465	2.366
AN	# AE1 N #	7.127	2.376
ANNE	# AE1 N #	7.127	2.376
ANN	# AE1 N #	7.127	2.376
FOREIGN	# F AO1 R AH0 N #	14.522	2.420
FORE	# F AO1 R #	9.681	2.420
FOR	# F AO1 R #	9.681	2.420
FOUR	# F AO1 R #	9.681	2.420
FAURE	# F AO1 R #	9.681	2.420
THAT(2)	# DH AH0 T #	9.699	2.425
WAS	# W AA1 Z #	9.812	2.453
AND(2)	# AH0 N D #	9.853	2.463
WAND	# W AA1 N D #	12.321	2.464
STU	# S T UW1 #	9.890	2.473
STEW	# S T UW1 #	9.890	2.473
HAS	# HH AE1 Z #	10.061	2.515
AUNT	# AE1 N T #	10.063	2.516
ANT	# AE1 N T #	10.063	2.516
HAN'S(2)	# HH AE1 N Z #	12.642	2.528
HANS(2)	# HH AE1 N Z #	12.642	2.528
HANNES	# HH AE1 N Z #	12.642	2.528
WIND(2)	# W IH1 N D #	12.693	2.539
WEND	# W EH1 N D #	12.700	2.540
HAT	# HH AE1 T #	10.174	2.544
HANDS	# HH AE1 N D Z #	15.318	2.553
STATIONED	# S T EY1 SH AH0 N D #	20.498	2.562
WOULD	# W UH1 D #	10.301	2.575

Table 1: the good end of the English word list.

We see that the words in Table 2 barely look like English words at all: they consist of borrowings (from Hawaiian, Arabic, Hungarian, etc.) and sound-symbolic forms (like “yeah”). Needless to say, the Arabic words would not end up on the bottom of a list if the analysis had been derived from a corpus of Arabic – but this is the English phonological structure that is concerned. Bear in mind this ranking is by phonological analysis, and is *not* (for example) based on word frequency (except in an indirect way).

In Tables 3 and 4, we see the parallel forms from a dictionary of Japanese.[4] For Japanese, I had no information about word-frequencies (which I did in the case of English), so those effects do not enter into the results in Japanese.



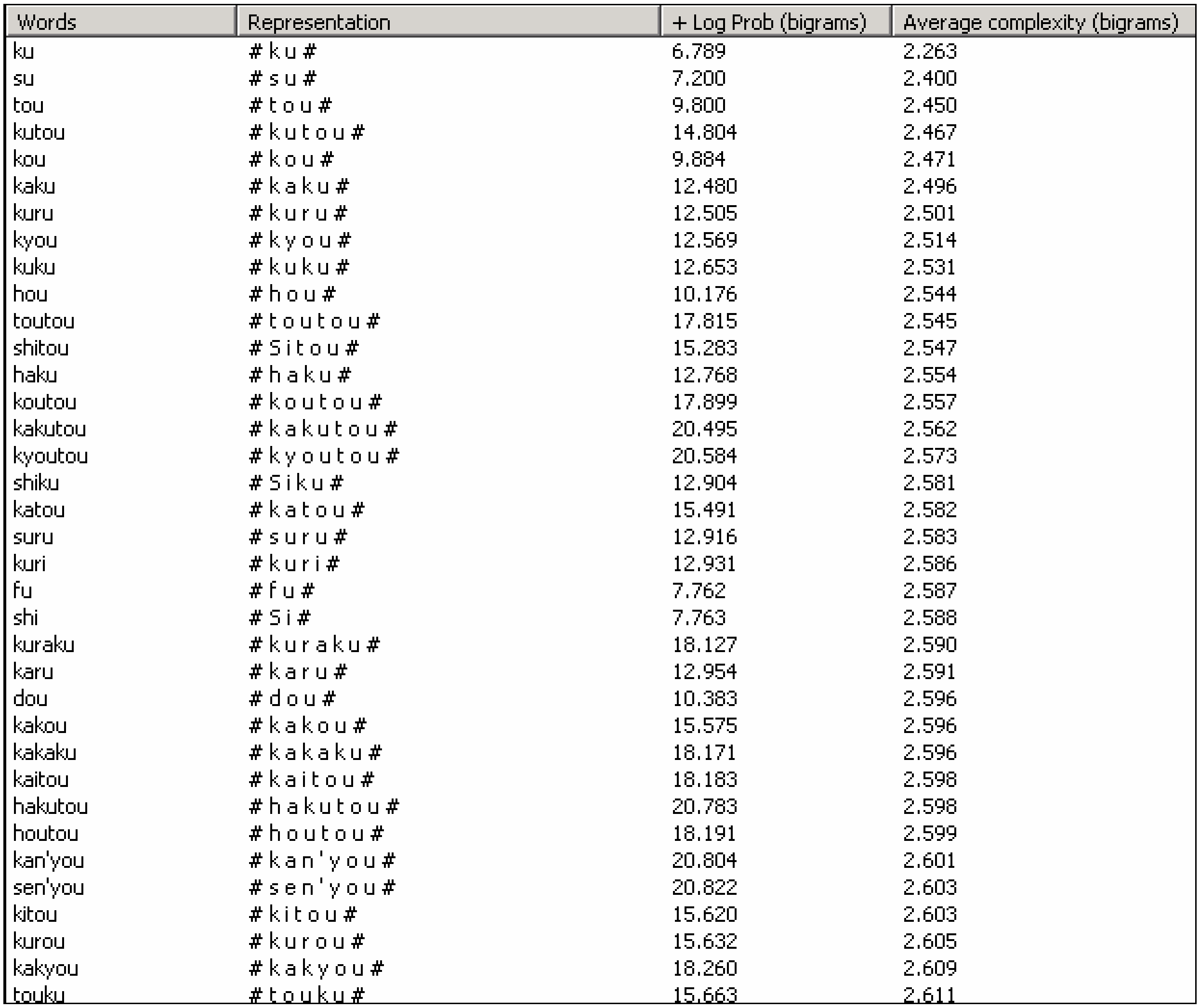
Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)
ZIYANG(2)	# ZH IY0 AA1 NG #	39.483	7.897
VOIGT	# V OY1 G T #	39.514	7.903
JIANGSU	# JH Y AA1 NG S UW0 #	55.703	7.958
THE(2)	# DH AH1 #	23.903	7.968
UDO	# UW1 D OW1 #	31.943	7.986
ZAYRE	# Z EY1 R #	31.985	7.996
OOMPH	# UW1 M F #	32.012	8.003
ZHUHAI	# Z UW1 HH AY1 #	40.043	8.009
ARAU	# AH0 R AW1 #	32.177	8.044
BOLSHOI	# B OW0 L SH OY1 #	48.375	8.062
ATSUSHI	# AA0 S S UW0 SH IY0 #	56.483	8.069
NIIHAU	# N IY1 HH AW0 #	40.434	8.087
OOH	# UW1 #	16.231	8.115
LITHGOW	# L IH1 TH G AW0 #	48.877	8.146
L'HEUREUX	# L HH Y UW1 R UH1 #	57.082	8.155
OOLONG	# UW1 L AO0 NG #	40.821	8.164
MUI	# M UW1 IH0 #	32.746	8.187
ZHANG	# ZH AE1 NG #	32.777	8.194
ZWEIG	# Z W AY1 G #	41.038	8.208
UH	# AH1 #	16.605	8.303
ZAIRE	# Z AY0 IH1 R #	41.661	8.332
ZHIVKOV	# ZH IH1 V K AA0 V #	58.602	8.372
AER(2)	# EY1 IY1 AA1 R #	41.910	8.382
ZOE	# Z OW1 IY0 #	34.330	8.583
ZULAUF	# Z UW1 L AW0 F #	51.820	8.637
KUKJE	# K UW1 K Y IH0 #	52.041	8.674
YEAH	# Y AE1 #	26.086	8.695
SALEH	# S AA1 L EH0 HH #	52.990	8.832
ARROYO	# ER0 OY1 OW0 #	35.418	8.855
AI(2)	# EY1 AY1 #	26.833	8.944
DES(2)	# D IH1 #	27.197	9.066
EH	# EH1 #	18.149	9.075
OAHU	# OW1 AA1 HH UW0 #	46.047	9.209
ZHAO	# ZH AW1 #	27.758	9.253
CMU	# List #	19.950	9.975
ZSA	# ZH AA1 #	30.607	10.202

Table 2: "Bad" end of the English word list

And in Table 4, we see the "bad" end of the approximately 50,000 word Japanese list.[5]

The tables presented here are selections from the entire lists; the English list is over 100,000 words, and the Japanese is over 50,000 words; the tables given here are just the top and the bottom ends of these lists. As we can see, the "bad" ends of the lists contain primarily borrowings into the language, compounds, and onomatopoeia, while the "good" ends of the lists contain words whose phonological patterns are the most central in the language.

These lists rank the words of each lexicon by their average *probability*, and it is the character of probabilistic models (primarily in phonology) which I wish to discuss in this paper.

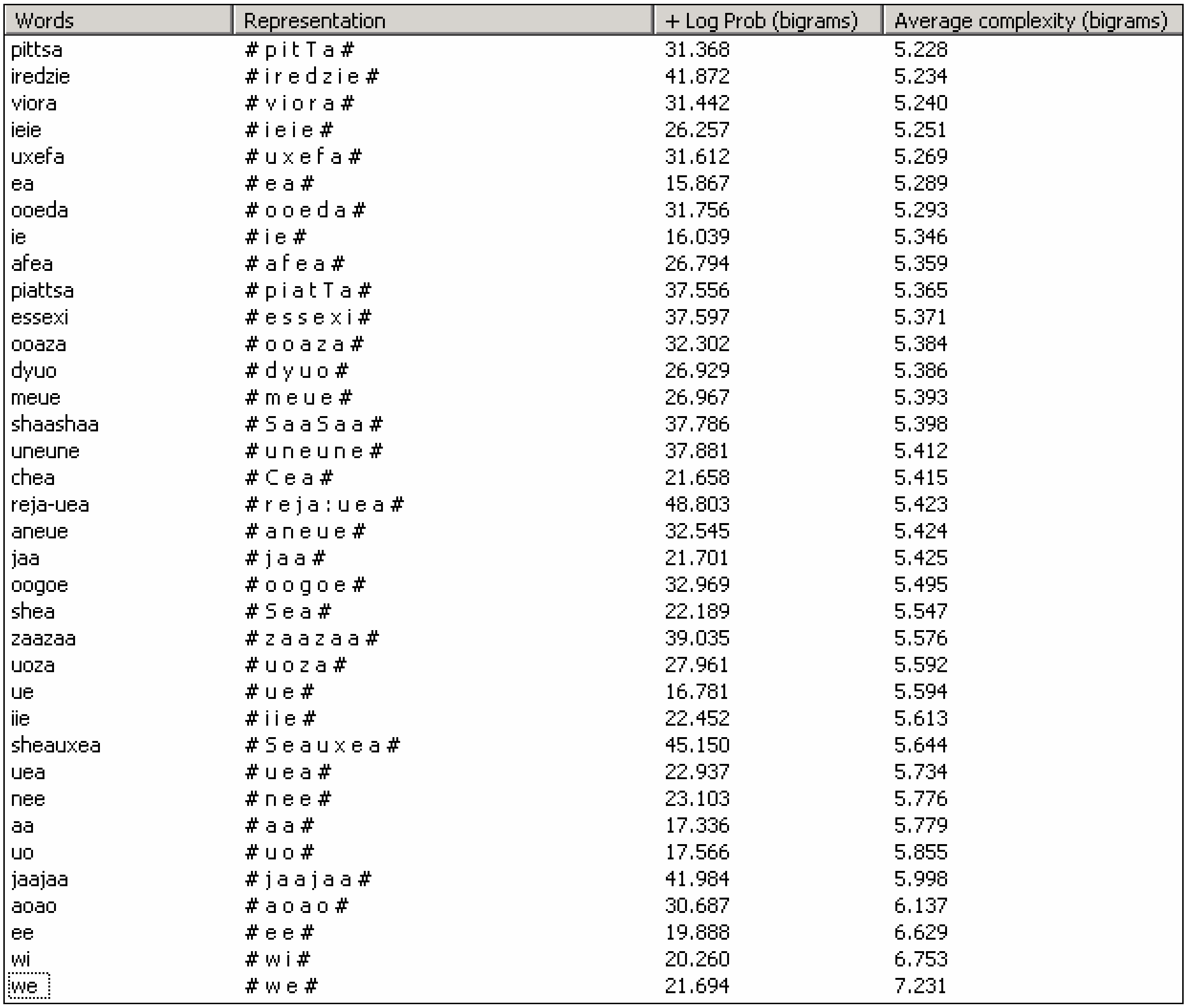


Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)
ku	# k u #	6.789	2.263
su	# s u #	7.200	2.400
tou	# t o u #	9.800	2.450
kutou	# k u t o u #	14.804	2.467
kou	# k o u #	9.884	2.471
kaku	# k a k u #	12.480	2.496
kuru	# k u r u #	12.505	2.501
kyou	# k y o u #	12.569	2.514
kuku	# k u k u #	12.653	2.531
hou	# h o u #	10.176	2.544
toutou	# t o u t o u #	17.815	2.545
shitou	# S i t o u #	15.283	2.547
haku	# h a k u #	12.768	2.554
koutou	# k o u t o u #	17.899	2.557
kakutou	# k a k u t o u #	20.495	2.562
kyoutou	# k y o u t o u #	20.584	2.573
shiku	# S i k u #	12.904	2.581
katou	# k a t o u #	15.491	2.582
suru	# s u r u #	12.916	2.583
kuri	# k u r i #	12.931	2.586
fu	# f u #	7.762	2.587
shi	# S i #	7.763	2.588
kuraku	# k u r a k u #	18.127	2.590
karu	# k a r u #	12.954	2.591
dou	# d o u #	10.383	2.596
kakou	# k a k o u #	15.575	2.596
kakaku	# k a k a k u #	18.171	2.596
kaitou	# k a i t o u #	18.183	2.598
hakutou	# h a k u t o u #	20.783	2.598
houtou	# h o u t o u #	18.191	2.599
kan'you	# k a n ' y o u #	20.804	2.601
sen'you	# s e n ' y o u #	20.822	2.603
kitou	# k i t o u #	15.620	2.603
kurou	# k u r o u #	15.632	2.605
kakyou	# k a k y o u #	18.260	2.609
touku	# t o u k u #	15.663	2.611

Table 3 “Good” end of the Japanese list

Probability theory is not very well known to linguists, or to the educated public, in general. We all know that the probability of rolling a die and getting a 3 is 1 out of 6, or slightly more than 16%, and we probably all know that the probability of tossing a coin 3 times, and getting *heads* all three times is 1/8, because 1/8 is ½ times ½ times ½. But this knowledge (although it is correct) is misleading in the long run, because it tends to suggest that the goal of probability theory is to determine precisely how rare a particular outcome of a random event is. Since phonologists know that their concern is not with particularly rare events, and certainly not with random events, it is not at all obvious to phonologists at first why they should have any interest at all in probabilistic models. The bottom line is this: the theory of probability is fundamentally the quantitative theory of evidence.

And yet the words in Tables 1 and 2, and 3 and 4, are the words of English and Japanese ranked by their (average) *probability* – one model of their phonological probability – and it is perfectly clear from looking at the list that this is not a list of the frequencies of words in English and Japanese. So what is it? My purpose in this paper is to explain what is involved here, and why we as phonologists might be interested in this kind of modeling.



Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)
pittsa	# p i t T a #	31.368	5.228
iredzie	# i r e d z i e #	41.872	5.234
viora	# v i o r a #	31.442	5.240
ieie	# i e i e #	26.257	5.251
uxefa	# u x e f a #	31.612	5.269
ea	# e a #	15.867	5.289
ooeda	# o o e d a #	31.756	5.293
ie	# i e #	16.039	5.346
afea	# a f e a #	26.794	5.359
piattsa	# p i a t T a #	37.556	5.365
essexi	# e s s e x i #	37.597	5.371
ooaza	# o o a z a #	32.302	5.384
dyuo	# d y u o #	26.929	5.386
meue	# m e u e #	26.967	5.393
shaashaa	# S a a S a a #	37.786	5.398
uneune	# u n e u n e #	37.881	5.412
chea	# C e a #	21.658	5.415
reja-uea	# r e j a : u e a #	48.803	5.423
aneue	# a n e u e #	32.545	5.424
jaa	# j a a #	21.701	5.425
oogoe	# o o g o e #	32.969	5.495
shea	# S e a #	22.189	5.547
zaazaa	# z a a z a a #	39.035	5.576
uoza	# u o z a #	27.961	5.592
ue	# u e #	16.781	5.594
iie	# i i e #	22.452	5.613
sheauxea	# S e a u x e a #	45.150	5.644
uea	# u e a #	22.937	5.734
nee	# n e e #	23.103	5.776
aa	# a a #	17.336	5.779
uo	# u o #	17.566	5.855
jaajaa	# j a a j a a #	41.984	5.998
aoao	# a o a o #	30.687	6.137
ee	# e e #	19.888	6.629
wi	# w i #	20.260	6.753
we	# w e #	21.694	7.231

Table 4: "Bad" end of the Japanese word list

My own interest in this approach goes somewhat deeper than an interest in a different kind of modeling. I believe that a probabilistic approach to linguistic modeling allows us to defend a very different conception of what linguistics is: in particular, this is the view that grammars are scientific models of linguistic data, rather than models of what exists in the heads of speakers. In the end, I believe that mainstream linguistics is *not* a cognitive science in the *same* sense that psychology is, although this does not lessen the interest of linguistics to psychologists who study the human use of language. Linguistics is the science that addresses linguistic data; it is of interest to us because language is a peculiarly human and symbolic activity, certainly; but linguistics teaches us to deal with language, not with organs such as brains.[6]

One of the central notions that we will explore – and this is not a new idea, in some domains – is this: when we wish to analyze the phonology of a language, we establish a sizeable set of observations O ("corpus") of the language, and we attempt to build a model M that maximizes the probability of that set of observations O. The model M is a model of the phonology of the language; once it is constructed, it is capable of assigning a probability to *any* set of observations

O′ from the language. The probability that M assigns to any particular set of observations will be extremely small, and the larger the set of observations is, the smaller is its probability. I emphasize this from the start; the fact that a particular utterance has a very low probability is not a problem in *any* sense, and probability cannot be said to be the same thing as (un)grammaticality. To repeat, the goal is to develop a model of phonology which assigns probabilities, and in particular to find the phonological model which assigns the *highest* probability to the set of observed data. This probability need not be large, but the ultimate claim is that if we state the goal of the theory in this way, the model M which does assign the highest probability to the observed data will be the best linguistic model of the observed data. But it will take us some time to explain why this should be the case.[7]

One last point to mention at the beginning: the development of probabilistic models or grammars is not an activity that can, in practical terms, be carried out by a human being unaided by computer. What one must do is to figure out how one's phonological idea can be expressed in algorithmic terms – and then that idea must be turned into a computer program, which we will call an *abstract model*. This abstract model, the embodiment of the phonological idea as a computer program, is capable of taking as input a large set of phonological data and creating a specific model of that data (which I will call an *instance* of the model); and, more to the point, it is capable of developing an instance of the model of *any* set of observations. The instance of the model can then be used to analyze further data that we present to it. In short, an (abstract) model that we come up with from the probabilistic point of view is always at a higher level of generalization than the kinds of ideas that we tend to develop in familiar generative terms. A probabilistic (abstract) model will always be capable of accepting many different sets of data and analyzing them in self-consistent ways.

2. Introduction to probability

One of the most enlightening aspects of probabilistic models of phonology and morphology is that building such models forces the linguist to think explicitly about questions that one formerly took for granted, with little conscious reflection, in building non-probabilistic models. In a sense, we can say that a probabilistic model consists of a non-probabilistic model plus some numerical quantities; it is *not* true that probabilistic models are inherently simpler or less structural than non-probabilistic models.

The two essential characteristics of a probabilistic model are these: (1) it must define and characterize the entire universe of possible events (or observations) in the domain that it models – for us, this might be all possible utterances in a language L; and (2) it must assign a number (which we call a "probability") to each of those events (or utterances): these probabilities are all non-negative (that is, zero or positive) -- and the probabilities must add up to 1.0, when we add up the probabilities of the entire domain. The larger the universe of possible utterances, the smaller the average probability is going to be (roughly), if they must all, taken together, add up to 1.0.[8] The universe of possible utterances is generally called the *sample space* in probability theory, and a set of non-negative numbers that add up to 1.0 is called a *distribution*.

This much, so far, is essentially mathematical, and not scientific. When we look at a model of a real system – such as a phonological model – there is another important characteristic: the probability that we assign to each utterance (that is, to each event in the sample space) is assigned by a mathematical function that we devise ourselves which is built up out of properties of the subpieces of the utterance. If we are building a phonological model, then the probability of the utterance will be built up out of properties of the phonological subpieces, such as the phonemes, their order, and their phonological organization, that is, what we call the phonological representation. (We might also expect to employ notions like distinctive features, syllable structure, sonority, etc.) So our goal will be to figure out how phonological representation can contribute to assigning probabilities!

Here is perhaps the crucial point. When we build our probabilistic model, we assign probabilities to the small subpieces of the model (for example, probabilities of individual phonemes or features), and these probabilities are usually tightly linked to direct observation (roughly, but only roughly, we take these probabilities to be equal to their observed frequencies). In the case at hand, this might be the probabilities of particular phonemes. Then our model assigns probabilities to the larger pieces – for example, to words – based on two things: (i) our theoretical (abstract) model and (ii) the probabilities now assigned to the more elementary units (the instance of the model).

Let us begin with an extremely elementary model, the *unigram* model, the model that assumes that individual phonemes can be assigned a particular probability, and that the probability that we will assign to a sequence of phonemes is the product of the probabilities of the individual phonemes. In addition, we assume that all words end with a particular symbol (#) marking word-end.

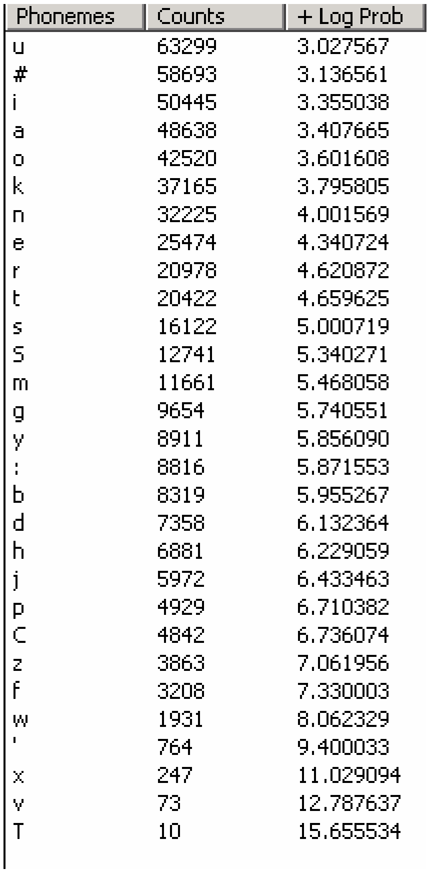
It is important to keep track of the fact that we are already talking about two distinct (but closely related) distributions (and remember: a distribution is a set of non-negative numbers that add up to 1.0): the set of probabilities assigned to individual phonemes, and the set of probabilities that are assigned to all possible words in the language. These are two quite different universes, with no overlap: a word must contain at least one phoneme, and it must end with an instance of the end-of-word symbol # (and thus a phoneme can never be a word, in this model). But we can connect them by means of the unigram assumption that order makes no difference in computing probabilities, which is to say, the probability of a string is the product of the probabilities of the individual phonemes.

Mathematically, it is much more convenient to think about the *logarithm* (or *log*, for short) of the probabilities instead of the probabilities themselves. Permit me to remind the reader that a logarithm of x, base 2, is the exponent to which one must raise 2 in order to get x: $2^{\log_2 x} = x$ and $\log_2(2^x) = x$. $2^3 = 8$; therefore $\log_2 (8) = 3$. $2^7 = 128$, and therefore $\log_2 (128) = 7$. $2^{-1} = ½$, and therefore $\log_2(½) = -1$. $2^{-3} = 1/8$, and therefore $\log_2 (1/8) = -3$. Bear in mind that finding the log of x times y yields the same result as adding the log of x and the log of y.

Because a probability is always a number between 0 and 1, its logarithm will be negative or zero, and since most of us prefer positive numbers, it is traditional to talk about -1 * log of the probability, because this is a positive number (or non-negative, at least). I will refer to this as the *positive log probability*.[9]

In Table 5 will be found a chart of phonemes of Japanese with positive log probabilities computed from Breen's online dictionary of Japanese, based essentially on his romaji characterizations of words.

Why is the positive log probability more convenient than the probability itself? The answer is that log probabilities are added (rather than multiplied), and this makes things a good deal simpler. To calculate the log probability of a word under the simple unigram model, we add the log probabilities of the individual segments.[10] (Remember that the "unigram model" is defined as the model which assumes that the probability of a unit is independent of its context).Therefore we will prefer to look at the column in our tables which are labeled "positive log probability", and we will ask, What is the log probability of various words? Since the log probability of A times B is the sum of the log probability of A and the log probability of B, it follows that the log probability of a sequence of segments is the *sum* of the log probabilities of the individual segments. It also follows that we can easily calculate the *average log probability* for any given word: we compute the total log probability, and divide that by the number of segments in the word. This constitutes the phonological complexity of a word in that language, under that model. This quantity will be extremely important to us in what follows.



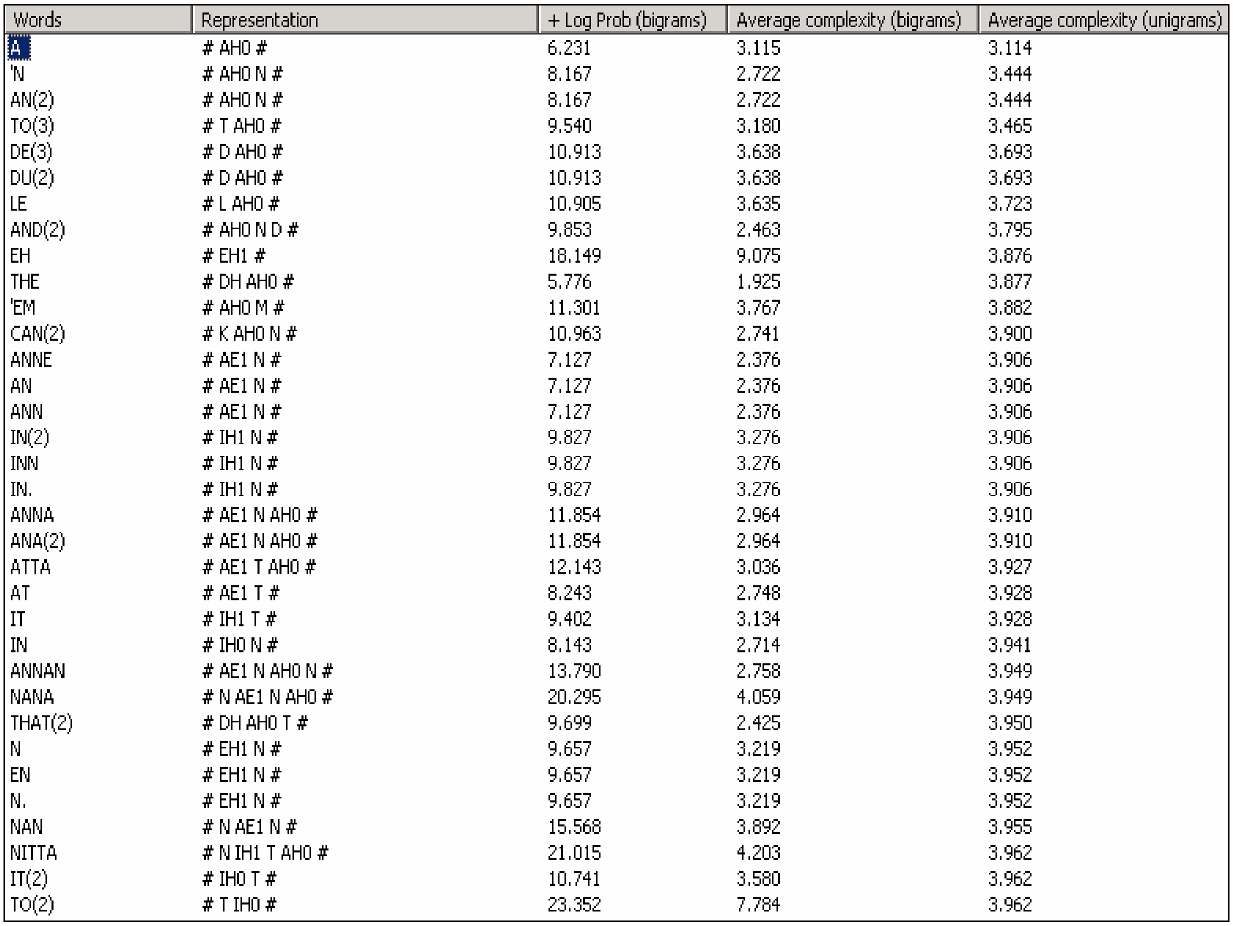
Phonemes	Counts	+ Log Prob
u	63299	3.027567
#	58693	3.136561
i	50445	3.355038
a	48638	3.407665
o	42520	3.601608
k	37165	3.795805
n	32225	4.001569
e	25474	4.340724
r	20978	4.620872
t	20422	4.659625
s	16122	5.000719
S	12741	5.340271
m	11661	5.468058
g	9654	5.740551
y	8911	5.856090
:	8816	5.871553
b	8319	5.955267
d	7358	6.132364
h	6881	6.229059
j	5972	6.433463
p	4929	6.710382
C	4842	6.736074
z	3863	7.061956
f	3208	7.330003
w	1931	8.062329
'	764	9.400033
x	247	11.029094
v	73	12.787637
T	10	15.655534

Table 5: Phonemes of Japanese

Does the phonologist have an understanding, or an intuition, regarding this quantity, the average log probability of a word? Not really, under the conditions which I have presented so far. Let us take a look at the probabilities assigned to words of English and Japanese using the unigram model (which assumes that phonemes have no relationship to the phonemes that are next to them). We do this by computing the probability of each word in our corpus, and then ranking words by their average log probability.

We find a ranking as given in Tables 6 – 9: Tables 6 and 7 for English, and Tables 8 and 9 for Japanese.

What do we notice? Two things strike us right away: first, the low-probability words (that is, high complexity words) at the "bad" end are indeed borrowings (they have a high proportion of unusual phonemes, after all), but the words at the good end are often strange words. In Japanese, it is quite striking: in the good list, we find a good selection of the odd words that consist entirely, or almost entirely, of vowels. Why should that be so? The answer is clear with a moment's thought: individual vowels are more common than consonants because there are fewer distinct vowels than there are distinct consonants, so the words that would score the best on this test will be those composed only of high-frequency phonemes, which tends to be the vowels. In English, we find a similar effect: we find at the “good” end just those words that are built entirely out of the highest-frequency phonemes – basically, schwa, /a/, and coronal consonants.

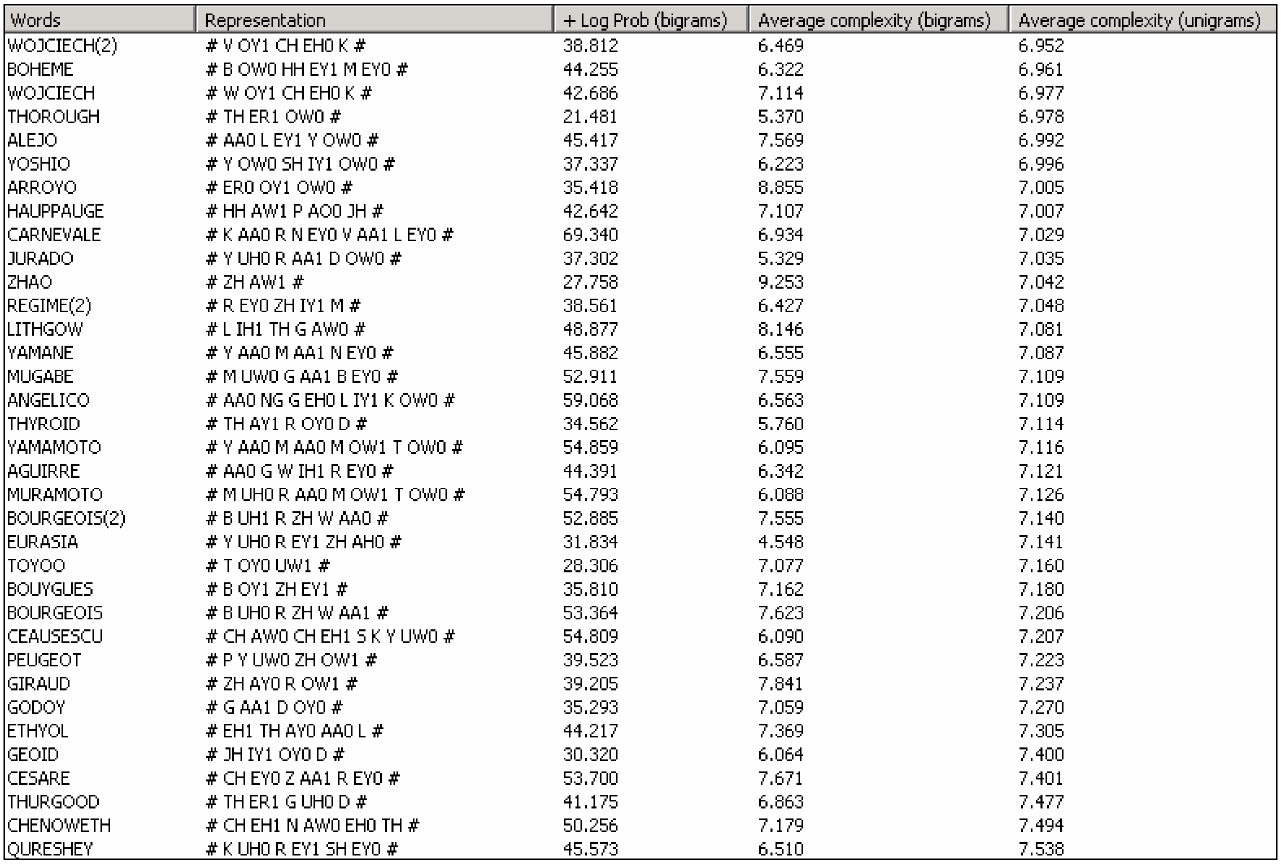


Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)	Average complexity (unigrams)
A	# AH0 #	6.231	3.115	3.114
'N	# AH0 N #	8.167	2.722	3.444
AN(2)	# AH0 N #	8.167	2.722	3.444
TO(3)	# T AH0 #	9.540	3.180	3.465
DE(3)	# D AH0 #	10.913	3.638	3.693
DU(2)	# D AH0 #	10.913	3.638	3.693
LE	# L AH0 #	10.905	3.635	3.723
AND(2)	# AH0 N D #	9.853	2.463	3.795
EH	# EH1 #	18.149	9.075	3.876
THE	# DH AH0 #	5.776	1.925	3.877
'EM	# AH0 M #	11.301	3.767	3.882
CAN(2)	# K AH0 N #	10.963	2.741	3.900
ANNE	# AE1 N #	7.127	2.376	3.906
AN	# AE1 N #	7.127	2.376	3.906
ANN	# AE1 N #	7.127	2.376	3.906
IN(2)	# IH1 N #	9.827	3.276	3.906
INN	# IH1 N #	9.827	3.276	3.906
IN.	# IH1 N #	9.827	3.276	3.906
ANNA	# AE1 N AH0 #	11.854	2.964	3.910
ANA(2)	# AE1 N AH0 #	11.854	2.964	3.910
ATTA	# AE1 T AH0 #	12.143	3.036	3.927
AT	# AE1 T #	8.243	2.748	3.928
IT	# IH1 T #	9.402	3.134	3.928
IN	# IH0 N #	8.143	2.714	3.941
ANNAN	# AE1 N AH0 N #	13.790	2.758	3.949
NANA	# N AE1 N AH0 #	20.295	4.059	3.949
THAT(2)	# DH AH0 T #	9.699	2.425	3.950
N	# EH1 N #	9.657	3.219	3.952
EN	# EH1 N #	9.657	3.219	3.952
N.	# EH1 N #	9.657	3.219	3.952
NAN	# N AE1 N #	15.568	3.892	3.955
NITTA	# N IH1 T AH0 #	21.015	4.203	3.962
IT(2)	# IH0 T #	10.741	3.580	3.962
TO(2)	# T IH0 #	23.352	7.784	3.962

Table 6: “Good” end of unigram-ranked list: English

This model does not assign enough probability to words with natural sequences of phonemes; after all, it was built based on the assumption that phonemes are irrelevant to their neighbors. How can we take sequences into account?

The natural way to do that is to consider the probability of a phoneme as being dependent on the phoneme that precedes it.[11] We compute the probability of each phoneme, given the preceding phoneme. This is the part that is empirically driven; we then use a new model to compute the probability of each word: we say that the probability of a word is the product of the probability of each phoneme, conditioned by the preceding phoneme. We write this in the following way. We use the bracket notation S[i], where S is any string, to indicate the symbol in the ith place. If W = pit, then W[3] = t. We shall have calculated (and observed) the frequencies of a given phoneme P, if it immediately follows phoneme Q, and we express that (in a way that is comfortable to phonologists) as prob (P | Q _). This may be read as "the probability of P, given that the preceding phoneme is Q".



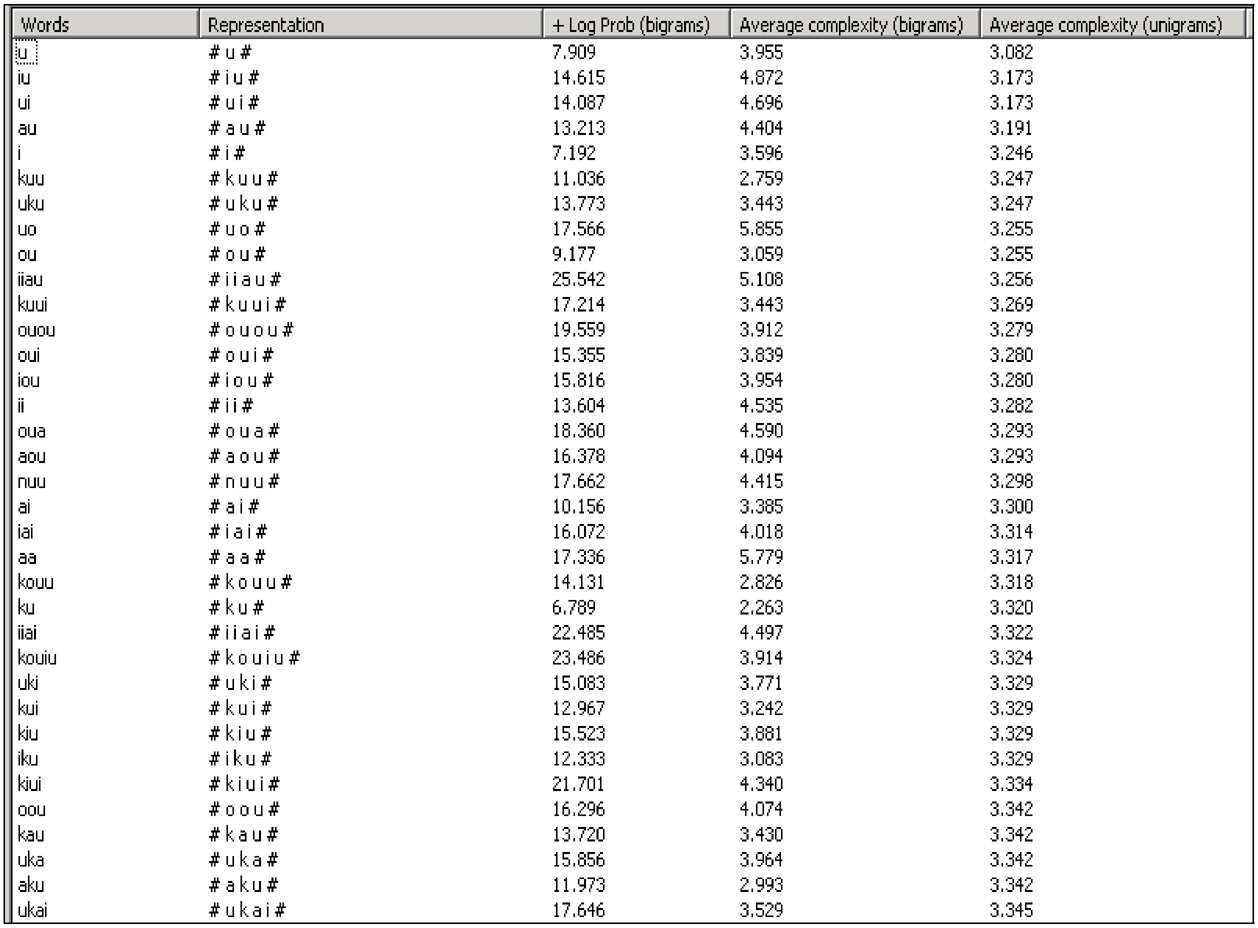
Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)	Average complexity (unigrams)
WOJCIECH(2)	# V OY1 CH EH0 K #	38.812	6.469	6.952
BOHEME	# B OW0 HH EY1 M EY0 #	44.255	6.322	6.961
WOJCIECH	# W OY1 CH EH0 K #	42.686	7.114	6.977
THOROUGH	# TH ER1 OW0 #	21.481	5.370	6.978
ALEJO	# AA0 L EY1 Y OW0 #	45.417	7.569	6.992
YOSHIO	# Y OW0 SH IY1 OW0 #	37.337	6.223	6.996
ARROYO	# ER0 OY1 OW0 #	35.418	8.855	7.005
HAUPPAUGE	# HH AW1 P AO0 JH #	42.642	7.107	7.007
CARNEVALE	# K AA0 R N EY0 V AA1 L EY0 #	69.340	6.934	7.029
JURADO	# Y UH0 R AA1 D OW0 #	37.302	5.329	7.035
ZHAO	# ZH AW1 #	27.758	9.253	7.042
REGIME(2)	# R EY0 ZH IY1 M #	38.561	6.427	7.048
LITHGOW	# L IH1 TH G AW0 #	48.877	8.146	7.081
YAMANE	# Y AA0 M AA1 N EY0 #	45.882	6.555	7.087
MUGABE	# M UW0 G AA1 B EY0 #	52.911	7.559	7.109
ANGELICO	# AA0 NG G EH0 L IY1 K OW0 #	59.068	6.563	7.109
THYROID	# TH AY1 R OY0 D #	34.562	5.760	7.114
YAMAMOTO	# Y AA0 M AA0 M OW1 T OW0 #	54.859	6.095	7.116
AGUIRRE	# AA0 G W IH1 R EY0 #	44.391	6.342	7.121
MURAMOTO	# M UH0 R AA0 M OW1 T OW0 #	54.793	6.088	7.126
BOURGEOIS(2)	# B UH1 R ZH W AA0 #	52.885	7.555	7.140
EURASIA	# Y UH0 R EY1 ZH AH0 #	31.834	4.548	7.141
TOYOO	# T OY0 UW1 #	28.306	7.077	7.160
BOUYGUES	# B OY1 ZH EY1 #	35.810	7.162	7.180
BOURGEOIS	# B UH0 R ZH W AA1 #	53.364	7.623	7.206
CEAUSESCU	# CH AW0 CH EH1 S K Y UW0 #	54.809	6.090	7.207
PEUGEOT	# P Y UW0 ZH OW1 #	39.523	6.587	7.223
GIRAUD	# ZH AY0 R OW1 #	39.205	7.841	7.237
GODOY	# G AA1 D OY0 #	35.293	7.059	7.270
ETHYOL	# EH1 TH AY0 AA0 L #	44.217	7.369	7.305
GEOID	# JH IY1 OY0 D #	30.320	6.064	7.400
CESARE	# CH EY0 Z AA1 R EY0 #	53.700	7.671	7.401
THURGOOD	# TH ER1 G UH0 D #	41.175	6.863	7.477
CHENOWETH	# CH EH1 N AW0 EH0 TH #	50.256	7.179	7.494
QURESHEY	# K UH0 R EY1 SH EY0 #	45.573	6.510	7.538

Table 7: "Bad" end of unigram-ranked list: English

Again, we compute those frequencies from a corpus, and use those frequencies as our probabilities, and compute the probability of each word as the product of all of these conditional probabilities:

$$\prod_{i=1}^{n=length(S)} prob(S[i] \mid S[i-1]\_\_)$$

This model is called the *bigram* model.[12] Let us assign log probability on the basis of this bigram model. As I indicated above, it is natural to look at the phonological complexity, i.e., the average log probability (rather than the total log probability) This is because phonological properties are, for the most part, *intensive*: if one puts two phonologically well-formed words together, the result is, by and large, just as well-formed phonologically (it is not twice as well-formed, in particular). If we now rank the words in a corpus by average log probability, we get the lists that we began with, Tables 1 through 4 above, which express quite well the phonologist's intuition regarding the phonological well-formedness of words.

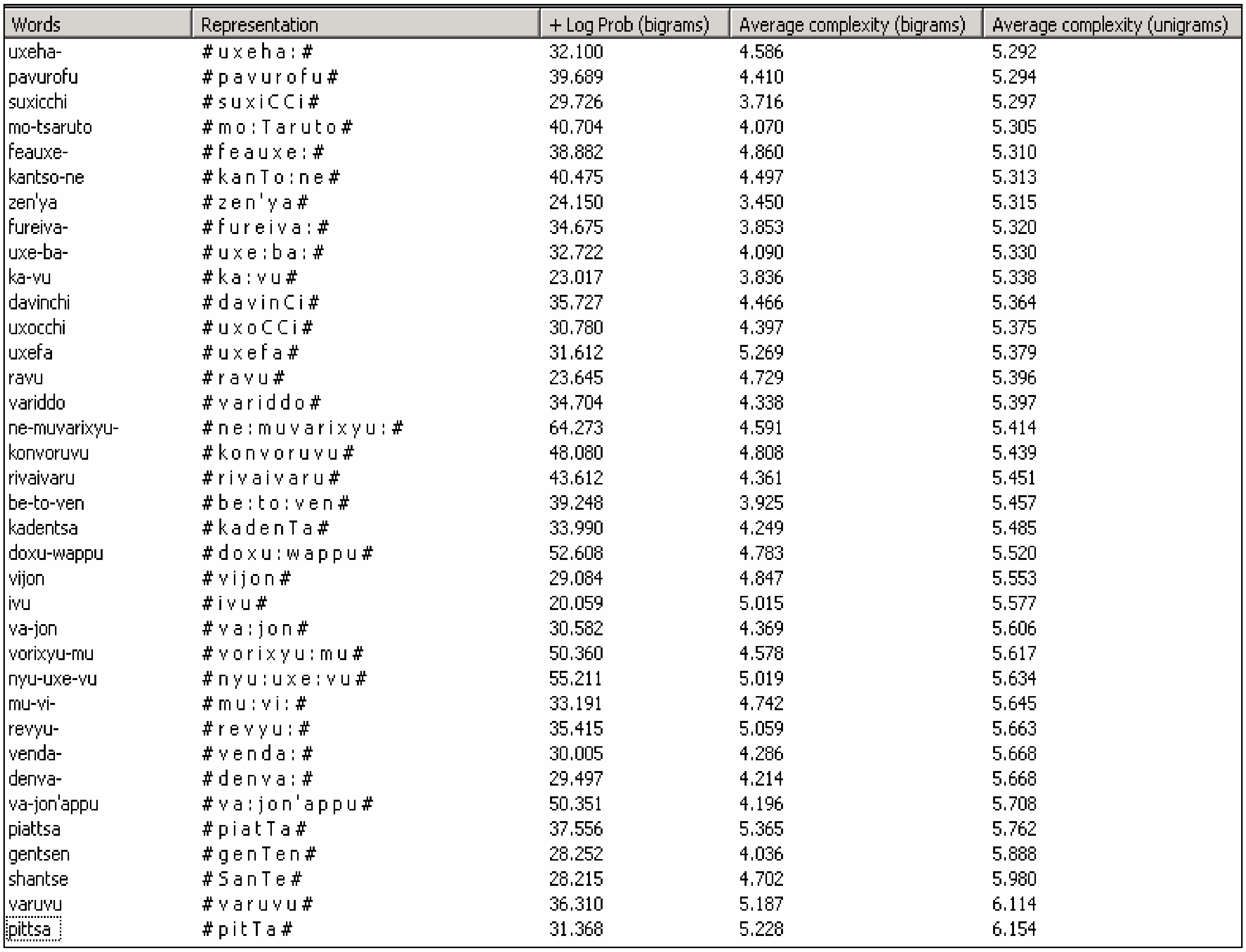


Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)	Average complexity (unigrams)
u	# u #	7.909	3.955	3.082
iu	# i u #	14.615	4.872	3.173
ui	# u i #	14.087	4.696	3.173
au	# a u #	13.213	4.404	3.191
i	# i #	7.192	3.596	3.246
kuu	# k u u #	11.036	2.759	3.247
uku	# u k u #	13.773	3.443	3.247
uo	# u o #	17.566	5.855	3.255
ou	# o u #	9.177	3.059	3.255
iiau	# i i a u #	25.542	5.108	3.256
kuui	# k u u i #	17.214	3.443	3.269
ouou	# o u o u #	19.559	3.912	3.279
oui	# o u i #	15.355	3.839	3.280
iou	# i o u #	15.816	3.954	3.280
ii	# i i #	13.604	4.535	3.282
oua	# o u a #	18.360	4.590	3.293
aou	# a o u #	16.378	4.094	3.293
nuu	# n u u #	17.662	4.415	3.298
ai	# a i #	10.156	3.385	3.300
iai	# i a i #	16.072	4.018	3.314
aa	# a a #	17.336	5.779	3.317
kouu	# k o u u #	14.131	2.826	3.318
ku	# k u #	6.789	2.263	3.320
iiai	# i i a i #	22.485	4.497	3.322
kouiu	# k o u i u #	23.486	3.914	3.324
uki	# u k i #	15.083	3.771	3.329
kui	# k u i #	12.967	3.242	3.329
kiu	# k i u #	15.523	3.881	3.329
iku	# i k u #	12.333	3.083	3.329
kiui	# k i u i #	21.701	4.340	3.334
oou	# o o u #	16.296	4.074	3.342
kau	# k a u #	13.720	3.430	3.342
uka	# u k a #	15.856	3.964	3.342
aku	# a k u #	11.973	2.993	3.342
ukai	# u k a i #	17.646	3.529	3.345

Table 8: “Good” end of unigram-ranked list: Japanese

We have now looked at two probabilistic models (unigram and bigram), and seen that each can generate a ranking of words from a corpus. The bigram-induced mapping is actually a surprisingly good automatic algorithm for determining how well any given individual word fits into the phonotactic patterns of the language. Borrowings will inevitably (I would suggest) consist of changes that bring words closer and closer to the heart of the phonology, which is to say, their nativization decreases their average log probability. (That is a conjecture, which should be tested empirically.)

There are a couple of striking differences between phonologists’ phonotactics and what we have just done here. First of all, the probability-based analysis does not distinguish between what is good or bad, in or out; it does produce a number, which can then be used to provide a ranking (words with lower average log probabilities outrank those with higher average log probability). Second, it has done this without reference to distinctive features or phonological classes (such as consonants and vowels). Third, the information that we (or the algorithm) have put together is rather verbose, or to put it differently, is distributed across a wide range of statistics. If in a language there is a very strong tendency for vowels and consonants to alternate, that knowledge is distributed across a wide range of statistics (that *t* follows *a* more than it follows *s*, etc.) As a linguist might say, the bigram model seems designed to miss the generalizations that phonologists hold dear.



Words	Representation	+ Log Prob (bigrams)	Average complexity (bigrams)	Average complexity (unigrams)
uxeha-	#uxeha:#	32.100	4.586	5.292
pavurofu	#pavurofu#	39.689	4.410	5.294
suxicchi	#suxiCCi#	29.726	3.716	5.297
mo-tsaruto	#mo:Taruto#	40.704	4.070	5.305
feauxe-	#feauxe:#	38.882	4.860	5.310
kantso-ne	#kanTo:ne#	40.475	4.497	5.313
zen'ya	#zen'ya#	24.150	3.450	5.315
fureiva-	#fureiva:#	34.675	3.853	5.320
uxe-ba-	#uxe:ba:#	32.722	4.090	5.330
ka-vu	#ka:vu#	23.017	3.836	5.338
davinchi	#davinCi#	35.727	4.466	5.364
uxocchi	#uxoCCi#	30.780	4.397	5.375
uxefa	#uxefa#	31.612	5.269	5.379
ravu	#ravu#	23.645	4.729	5.396
variddo	#variddo#	34.704	4.338	5.397
ne-muvarixyu-	#ne:muvarixyu:#	64.273	4.591	5.414
konvoruvu	#konvoruvu#	48.080	4.808	5.439
rivaivaru	#rivaivaru#	43.612	4.361	5.451
be-to-ven	#be:to:ven#	39.248	3.925	5.457
kadentsa	#kadenTa#	33.990	4.249	5.485
doxu-wappu	#doxu:wappu#	52.608	4.783	5.520
vijon	#vijon#	29.084	4.847	5.553
ivu	#ivu#	20.059	5.015	5.577
va-jon	#va:jon#	30.582	4.369	5.606
vorixyu-mu	#vorixyu:mu#	50.360	4.578	5.617
nyu-uxe-vu	#nyu:uxe:vu#	55.211	5.019	5.634
mu-vi-	#mu:vi:#	33.191	4.742	5.645
revyu-	#revyu:#	35.415	5.059	5.663
venda-	#venda:#	30.005	4.286	5.668
denva-	#denva:#	29.497	4.214	5.668
va-jon'appu	#va:jon'appu#	50.351	4.196	5.708
piattsa	#piatTa#	37.556	5.365	5.762
gentsen	#genTen#	28.252	4.036	5.888
shantse	#SanTe#	28.215	4.702	5.980
varuvu	#varuvu#	36.310	5.187	6.114
pittsa	#pitTa#	31.368	5.228	6.154

Table 9: "Bad" end of unigram-ranked list: Japanese

I will briefly make some observations about these points. Yes, while this approach models phonotactics without setting an absolute barrier between the forms that do and those that do not violate the phonotactics, a careful study of the words in the language (any language) suggest that this is a healthier and more accurate characterization of the facts. Languages gather words like a sun gathers planets: some are closer in and some are further out, and the best we can do is measure the force that links the two.

Why has there been no mention of distinctive features or phoneme classes, such as consonants and vowels? This is not the result of bringing in probability; it is simply a result of the very simple model that we are exploring. We could just as easily conduct a probabilistic analysis using features. For ease of presentation, I have limited the model to discussion of atomic phonemes, but we could perfectly well, and just as easily, perform the same analysis with features instead of phonemes.

Why is the phonotactic information distributed across the various numbers that we have built up? Why are the results not summarized in a small number of easily written-out statements or formulas?

There are two answers here. The first we have just stated: when we specify the model in terms of features, we will have statements that look more familiar. But there is a second answer. Our segmental bigram model of Japanese tells us that " u # " and " k u " are two of the pairs of segments with the greatest mutual information (i.e., the greatest statistical "stickiness" between them). That *is* important information about the sound pattern of Japanese, and we surely do not want to lose it. (Some of that information would be extracted by the feature-based phonotactic, certainly, but not all of it.) To put it slightly differently, we can now found out what phoneme-patterning which is specific to individual phonemes we find; and much of it does *not* reduce simply to feature-based statements.

We must go back now and reconsider why it was that we made the transition from the unigram model to the bigram model. The reader will recall that we did that primarily because the unigram model ranked much too high the Japanese words consisting entirely of vowels: the unigram model has no ability to respond to (or to capture) sequential patterns that are important in languages.

But according to the theory of probabilistic grammars, that is *not* the right answer to the question as to why we should change models. The *better* answer that the theory of probabilistic grammars gives is actually very subtle, and difficult to believe at first (perhaps even a bit hard to understand). The central point is this: do *anything* necessary to increase the total probability assigned to the corpus of observations.

Maximize the probability of the observations.

I did not show it before, but we will do it now: we will see that the shift from the unigram model to the bigram model leads to an significant increase in the probability assigned to the corpus. And the most important point of all is this: the discovery of any significant regularity will always lead to an increase in the probability assigned to the observations (i.e., a decrease in complexity).

How do we compute the probability of all of the observations? We merely need to compute the positive log probability of the entire corpus, which is simply the sum of the positive log probabilities of the individual words. Maximizing the probability is equivalent to minimizing the positive log probability. So let us add up the positive log probabilities for all of the words in the unigram model, and again do it in the bigram model. What do we find?

For our corpus of English, the total difference (in terms of log probability) between the two models is: 323,896 (unigram log probability: 1,883,085; bigram log probability 1,559,194), which is a 17% improvement.

To recap: the total positive log probability of the corpus under the bigram model is *smaller* than the positive log probability of the corpus under the unigram; therefore, the probability of the data under the bigram model is *greater* than it is under the unigram model, and therefore we must prefer the bigram model.

These numbers are in a natural unit: they are in “bits,” as defined by information theory, and the difference is called the mutual information, notions to which we will return.

We know now

- how to *calculate the probability of a word given a particular probabilistic model of its language*. We also know
- that it is simpler to talk about the (positive) log probability of the word rather than its probability. Because the (positive) log probability involves multiplication by -1,
- *maximizing* the probability is equivalent to *minimizing* the log probability. And we know that
- we can usefully calculate the *average log probability* of a word, by dividing the log probability of the word by the number of segments in the word.

If we compute the log probability for all the words in a language (by adding up the log probability of all of the words in the language), and form the average by dividing by the total number of characters in the corpus, what we get is the *entropy* of the system. In addition, the positive log probability of a word is also known as the *optimal compressed length* of a word, under the given model (this becomes important in the context of Minimum Description Length work; see note 7).

3. Identifying the language from which a word is drawn

Let us consider a related question. Suppose one is given a word (let us say the word is *Mitsubishi*) and a particular set of languages (say, English and Japanese), and one is asked to determine which language the word comes from. How could one do it?

First of all, why would one care? There are both technical and theoretical reasons for caring to do so. First of all, one might want to be build a device that would identify the language of a word in a document – one might want to invoke the appropriate spell-checker of a word processor automatically, or invoke a machine-translation system automatically. More theoretically, one might be interested in how multilingual people can automatically switch their receptive grammars – the language that they are listening in, so to speak – to be able to understand a speaker. One might be interested in how people can identify sublanguages within a language: in English, this means distinguishing Latinate from Germanic words; in Japanese, Sino-Japanese forms from native Japanese forms, and so on. How can this be done? How can one distinguish vocabularies, within a language or across languages?

Because we have been developing a probabilistic system, the answer is very easy to obtain. In our calculations so far, we have been computing for an individual word W and language L, what is the probability of word W? Now, however, we want to ask, given an individual word W, what is the probability that it comes from language L? But that is an easy modification to make, because it is simply a matter of applying Bayes’ rule, and comparing across languages. Bayes’ rule tells us how to make this inversion. It tells us, in this case, that the probability that word W comes from language L is equal to the probability of word W, given language L (which is what we have just computed), times the probability of language L and divided by the probability of word W:

Bayes' rule:

$$p(A\,|\,B) = probability\ of\ \ A\ if\ \ B\ is\ true = \frac{prob(B\,|\,A)\,prob(A)}{prob(B)}$$

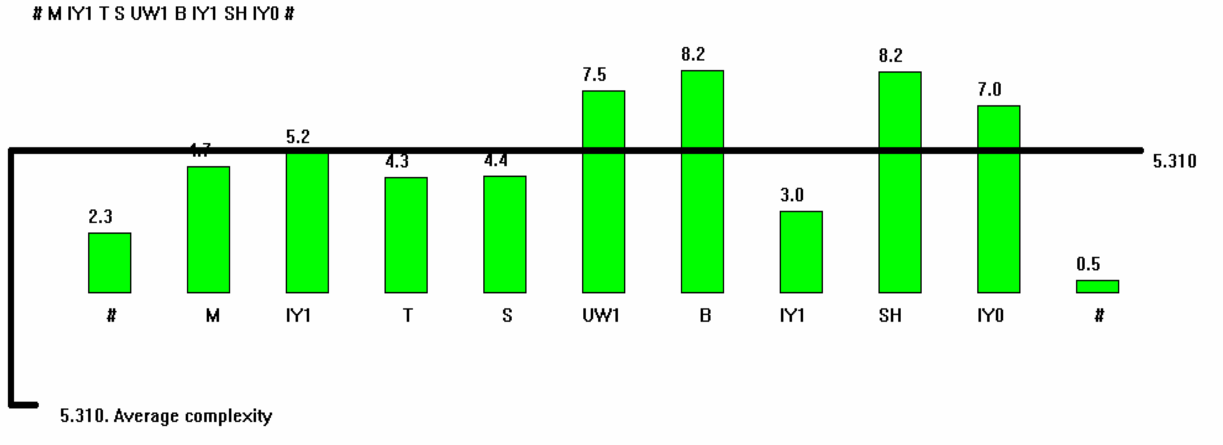
$$p(W\ comes\ from\ English\,|\,word\ W) = \frac{prob(word\ W\quad if\ W\ comes\ from\ English)\,prob(English)}{prob(word\ W)}$$

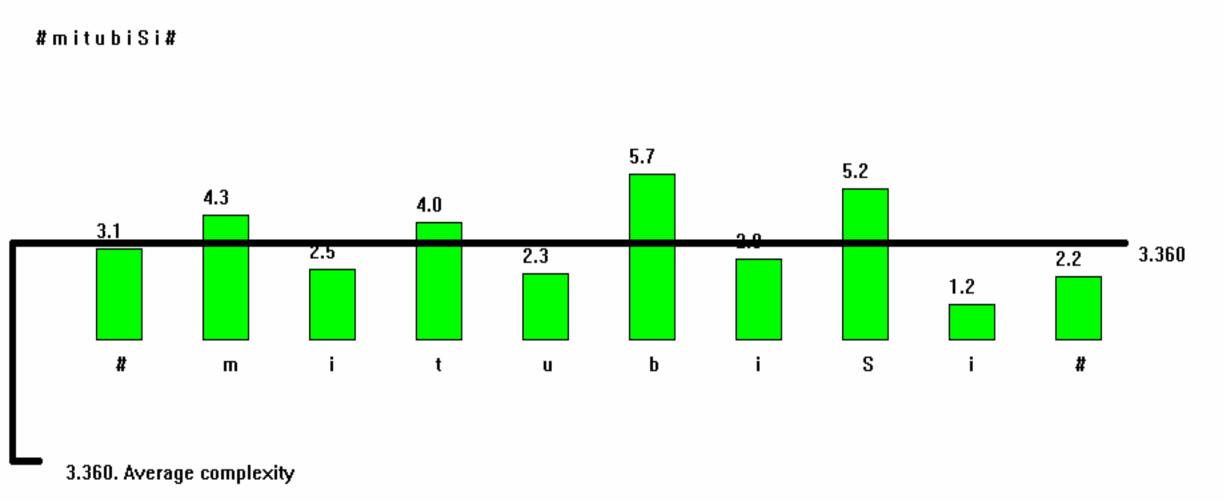
$$p(W\ comes\ from\ Japanese\,|\,word\ W) = \frac{prob(word\ W\quad if\ W\ comes\ from\ Japanese)\,prob(Japanese)}{prob(word\ W)}$$

So to solve the question as to which language a given word (such as *Mitsubishi*) comes from, we need to calculate this quantity for each language as shown above, and then see which probability is the greatest, and that will be the language that the word comes from – once again, it is a matter of *maximizing the probability of the evidence*.

What is the probability of a given language L (English or Japanese)? If we have no apriori reason to believe that one language is more likely than another, we may assign 0.50 to the probability of each language. What is the probability of the word *Mitsubishi* (or any other word)? This is hard to say, but in fact it does not matter, because we can ignore the denominator on the bottom, because whatever the value is, it will be the same for both languages, and thus our task becomes very simple: calculate the probability that word W comes from language L (which we already know how to do) for each language, and choose the language which assigns the highest probability.

If we compute the probability of *Mitsubishi* under the English model, we get an average log probability of 5.31; if we compute it under the Japanese model, we get 3.36. Therefore, *Mitsubishi* is a Japanese word.





Let us try it the other way around. Consider take the word *happen* (HH AE1 P AH0 N in English notation , h a p e n in Japanese). As Leibniz said, let us take out our pencils and let us calculate. The average log probability in English is 3.177, while in Japanese it is 3.434. Therefore, *happen* is an English word. And so on.

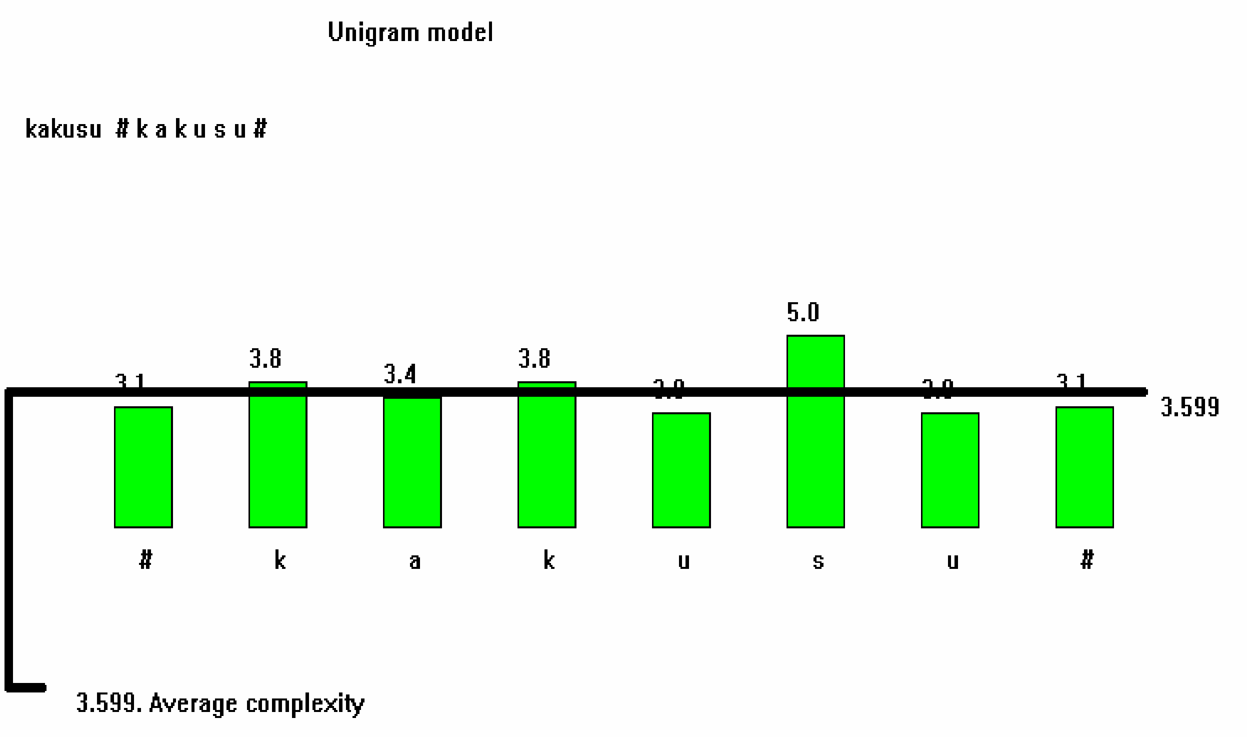
This is quite a remarkable result, and. I cannot imagine how another framework could accomplish such a result. In fact, the result is even more remarkable than I have indicated so far. If we have a string of words and we know that they all come from the same language, but we do not know (yet) which language it is, we can get the score for the entire string by adding together the scores of the individual words – so that a very strong indication from one word can overcome an incorrect score from a different word. That is, if we take the string *I bought a new Mitsubishi*, we will get a better English score than a Japanese score for the entire string, even though the Japanese score for the word *Mitsubishi* is better than its English score.[13]

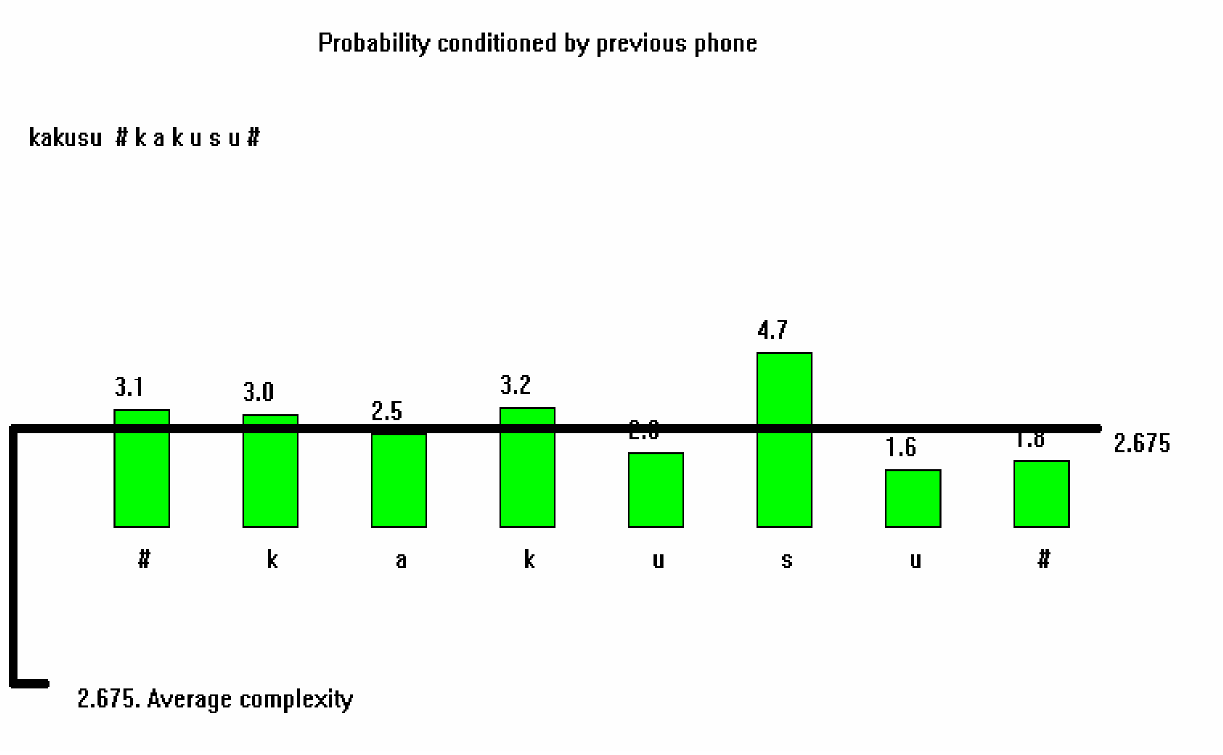
4. Other topics

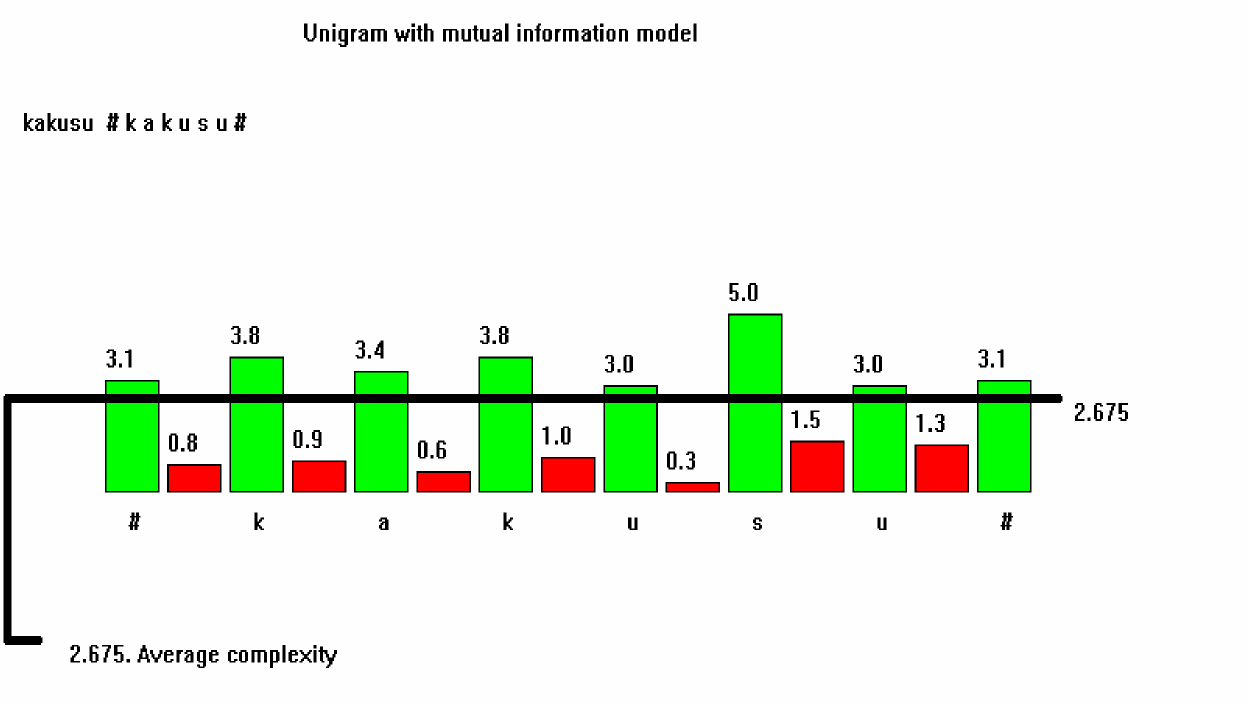
I would like to touch briefly on the following topics: (4.1) the notion of *mutual information*; the relation of this work to other approaches, such as (4.2) harmonic phonology, and (4.3) lexical phonology; (4.4) the relation of phonological representations (e.g., syllable structure, autosegmental structure) to probabilistic phonologies; (4.5) the extension of this work from phonotactics to morphophonemics, and (4.6) a remark on the relationship to optimality theory.

4.1 Mutual information

Let us look again at the program Complexity Sorter, and look at a wordlist from English. We have discussed the unigram and bigram models so far, and by clicking on the “1/2” button, we can switch back and forth between the two models. Now, if we compare the difference in average complexity for various words, one thing that we will discover is this: for words at the good end of the list (with the lowest complexity), the average complexity is lower when we look at the bigram complexity than when we look at the unigram complexity. That is, the model is a better predictor of a segment when it knows the segment that precedes. There is a name for the difference between the log probability based on the unigram model and the log probability based on the bigram model: it is *mutual information*, and we can display it by choosing the menu item “Model/unigram with MI”. When we select words near the good end of the list, the mutual information is virtually always positive (indicated here by a red rectangle). As we move towards the bad end of the list, where borrowings and other less-good words of the language appear, the mutual information gets smaller and smaller, and eventually turns negative, represented by blue rectangles hanging below the zero-line. These negative mutual informations represent the situation in which the two segments would really rather *not* be next to each other – they are phonotactically dis-preferred, as is typically found in borrowings, expressives, and so forth.







4.2 Harmonic phonology

I sketched an approach which I called harmonic phonology (Goldsmith 1990, 1993), which was based on the proposal that we could establish a measure of well-formedness computationally for any given phonological representation in a language. At the time that I wrote those earlier works, I did not know how to accomplish this task technically; the work described here today is the answer. At the time, I spoke of "maximizing the harmony" of a representation, but now it is clear that it is more convenient to speak of "minimizing the complexity, or information"; but whichever way we choose to describe it, the harmony is the log probability of the representation (and now, unlike above, when I say "log probability", I do *not* mean the positive log probability).

I conjectured in that early work that phonological rules apply if and only if their output is better-formed (in terms of complexity) than their input. This seems to me, now, an inappropriate suggestion, and I will return to an information-driven reformulation below.

4.3 Lexical phonology

It is probably not at all obvious, but some of the linguistic roots of what I have discussed here lie in lexical phonology.[14] Now, it is true that different people read lexical phonology in different ways: different people see the essence or the core of the theory in different ways. My reading of the theory (which I discussed in detail in Goldsmith 1990, Chapter 5) is this: the heart of the phonology is the lexical phonology, where the morphophonology lies; and in this component, all generalizations have a double-sided character: each generalization tells us (1) for each feature, what is the more likely value in a given phonological environment, and (2) for each feature (in a given environment), in what direction that feature may change if the language permits a change because of a derived environment, that is, because of a word-formation process. This is a remarkable notion, and it is one that has been adopted by many linguists, often, I think, without too much reflection. In some regards, it is adopted by the core of optimality theory, without much explicit discussion.

To repeat, according to lexical phonology, the lexical rules are both statements of probabilistic phonotactics and statements of allomorphy. How are rules of phonotactics and redundancy to be learned? And how much redundancy (i.e., patterning) must there be in the lexicon to make it "worthwhile" for the lexical phonology to set up a rule that accounts for an asymmetrical distribution of the values of a given feature? Phonologists have, to my knowledge, never addressed this question (though I posed it in Goldsmith 1995, admitting that I did not know how to pursue the question), but it is essential for making lexical phonology work. And the present work does precisely this: it says that all local redundancy is measured and captured.

4.4 The relation of phonological structure (or representation) to probabilistic phonology

I would like to repeat a point that I have already made: there is no intrinsic connection between probabilistic phonology and the extremely simple model of phonological structure that I have used so far in this talk, in which all structure is purely linear and there are no features. I have done that *only* for purposes of description and simplicity. In a probabilistic approach, articulated phonological structure is as important as it is in any other approach – indeed, more important, perhaps. When one develops a probabilistic model (not just in linguistics, but in any field) it is crucial for the analyzer to decide which factors may condition other factors in the model. In the bigram model that we looked at, we allowed neighbors to condition probabilities.

I believe that the real contribution of complex phonological representation is this: it allows us a richer idea of what it means for two items in a phonological representation to be "neighbors" – and it is only pairs and triples of neighbors that play a role in assigning probabilities (that is a conjecture).

4.5 About morphophonology

Constraints on space do not permit me to extend this discussion of phonotactics to a formulation of how to deal with morphophonology, but I would like to say a word about the matter.

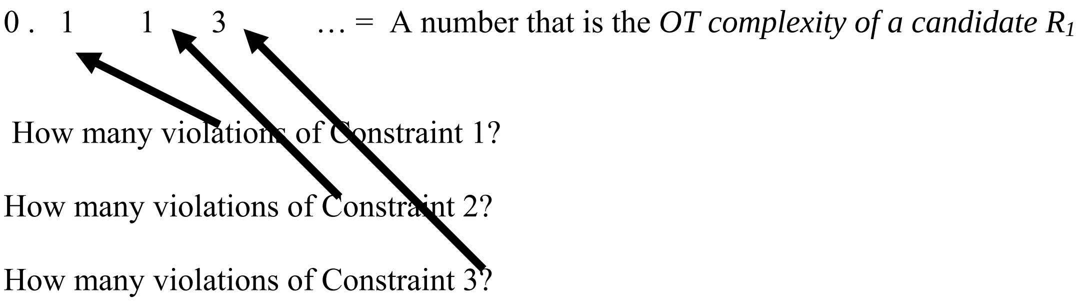
We have focused on the way in which language-particular complexity provides a mathematical mapping from representations to the real numbers. It follows that we can take a representation, and instead of keeping all of its elements fixed, we can let one (or more) of them vary across all of the possibilities in the language. For example, instead of computing the complexity of the string " # k l a b # ", I can make a variable out of the third position (let us indicate this as # k ? a b #; we can call that a *representation schema*), and then what I have is a function from all of the phonemes to the real numbers: for each phoneme P, I can replace "?" in "# k ? a b #" by P, and compute the complexity. We may then ask, which value of "?" gives us the *smallest* value for complexity? In that way, we can compute the *optimal* log probability of a representation-schema.

To develop a phonology with morphophonemics, we need to compute a two-level phonology (that is, a phonology with underlying forms and surface forms). Such a model contains, in essence, two phonological representations (one underlying, the other surface), with correspondences between elements on the two levels (as sketched in harmonic phonology, and many other phonologies). We compute the log probability of each of these links, across a training corpus. We use this information to compute the correct surface form SF, given a particular underlying form UF. For any given surface form SF, we compute the log probability of the pair (UF, SF), given SF – this computes the "reasonableness" of the pairing, corresponding to traditional phonological rules – and the log probability of the surface form SF, in the way that we have discussed today. We then choose the surface form for which the sum of these two values is the smallest (i.e., for which the probability is the greatest).

4.6 Optimality Theory

I would like to briefly consider the points of equivalence and of difference between a probabilistic approach and optimality theory. Let us briefly consider a notational variant of optimality theory that lends itself to a comparison with probabilistic models. We will call this variant "Weighted OT".

Consider an optimality theoretic ranking of a universal set of constraints. Assign a set of positive numbers (which we will call "weights") to these constraints in such a way that if a constraint C is ranked higher than constraint D, C's weight must be larger than D's weight. To make matters concrete, let us assign 0.1 to the highest-ranked constraint, 0.01 to the 2nd-ranked constraint, 0.001 to the 3rd constraint, and so on; the nth ranked constraint is assigned weight equal to 10^{-n}. For any given candidate phonological representation, we assign a score to it by adding up the number of times it violates each constraint in the hierarchy, which gives us the following picture; we may call the number (in between 0 and 1, by construction) generated by counting the constraint violations "the OT complexity":

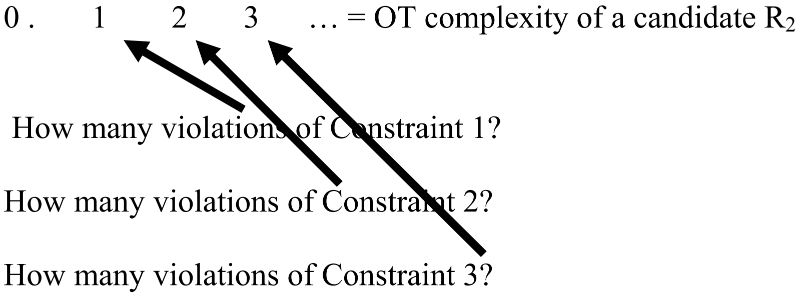


0 . 1 1 3 … = A number that is the *OT complexity of a candidate R_1*

How many violations of Constraint 1?

How many violations of Constraint 2?

How many violations of Constraint 3?



0 . 1 2 3 … = OT complexity of a candidate R_2

How many violations of Constraint 1?

How many violations of Constraint 2?

How many violations of Constraint 3?

	Constraint 1	Constraint 2	Constraint 3
☞Candidate R1	*	*	***
Candidate R2	*	**	***

In such a way, each candidate is assigned a number between 0 and 1 (here, 0.113 and 0.123), and classical optimality theory tells us to select the candidate with the smallest "OT complexity".

This is not the way OT is usually expressed, but a few moments' thought will convince the reader that this is so, and that choosing the candidate with the smallest OT complexity is essentially equivalent to working one's way through a tableau, looking for the surviving candidate.

There is a difference between the classical optimality tableau candidate selection algorithm and the rule that says "pick the candidate with the lowest OT complexity". The "OT complexity measure" described here proposes that there is a number (one less than the base of the number system used to express the number – here, base 10) such that you cannot count more than that number of violations. But since there is no preset limit on the base of the number system we will use, this claim has no significance.

There is no straightforward way to compare the actual *substance* of the constraints in OT and the elements being modeled probabilistically, but let us try to make such a comparison anyway. If we maintain the unrealistically simple unigram model of phonology, we can establish a simple parallelism between the (positive) log probability of a phoneme, on the one hand, and a constraint against that phoneme, on the other.

OT	Probabilistic model
Sample constraint: *s: it is assigned a rank in the hierarchy	"s" has a positive log probability: -1 * log prob(s) = 0.0021
Candidate with smallest OT complexity is selected	Candidate with small positive log probability is selected
Central premise: OT tableau mechanism + universal set of constraints	**Central premise**: maximize the probability of the observed data.

Probabilistic phonology and optimality theory can then be more easily compared. Both propose that candidate selection is an instance of minimization (hence, of optimization), but probabilistic phonology leaves no freedom regarding ranking or weighting of constraints: the weighting is directly established from the data, through assigning the positive log probability as the weight to each item in the model. In the probabilistic model, all pairs of adjacent items in a phonological representation can enter into the calculation of the probability. In its broadest sense, OT does not determine what may constitute a constraint; it offers only a means for adjudicating among conflicting constraints.

Notice that effects such as the "emergence of the unmarked" follow from a weight-based calculation; if two candidates are assigned the same weight by a more highly weighted constraint, it is a lower ranked and lower weighted constraint that will be decisive in determining the candidate with the optimal complexity.

5. Conclusion

What is a probabilistic phonology, as I have described it here? Is it a phonological theory, in the accepted sense of the term? One thing that it is *not* is a theory of how the mind works. But it offers a firm alternative foundation for phonology (and linguistics, more generally). It is not a generative account of phonology, and does not insist that rules be ordered in this way or that, or not at all; it is not like optimality theory, in proposing a specific algorithm for candidate selection and an innate inventory of constraints. It says only this: begin by expressing what one thinks are all of the conceivable events in the universe one wishes to describe. Consider a distribution across these events, which means assigning to each a probability in such a fashion that all of the probabilities sum to 1.0. The correct distribution is the one that maximizes the probability of the data which was described, most likely observed before the analysis was undertaken.

The general position that I have described here is often called positivism, and is characterized by a strong concern for observation and a great skepticism with regard to hypothetical objects whose plausibility derives from theory. Most of my life I have been dissatisfied with positivism, and I see no reason to change now. I do believe that scientific theories, under the best of circumstances, allow us to discover hidden realities behind or beyond the observed data. But my generation of linguists – those coming of age since mid-1960s – has become so deeply mired in anti-positivism that we have lost track of a good deal that is right and important about it. This is not the time or place to go into these matters at length, but I wish only to underscore the point that the position that I have argued for in these remarks is as much as anything a plea to return to a more balanced perspective regarding the relationship of evidence and theory in linguistics.[15]

I have suggested just a bit of what can be done with some elementary software which is freely available. I believe that there are things which we can learn about the phonology of a language by a careful inspection of the data that it presents to us, and I hope that the notions that I have discussed here may help in this task.

****

Notes

[1] I am grateful to Svetlana Soglasnova and Hisami Suzuki for discussions of the issues described here. Some of the suggestions made here have been influenced by the ongoing dissertation work by Svetlana Soglasnova concerning Russian hypocoristics, and by Daisuke Hara on American Sign Language; both develop detailed complexity measures of the systems they study, and go a good deal further than the remarks made in this paper.

[2] And see Solomonoff 1995 at http://world.std.com/~rjs/barc97.html

[3] There was a good deal of reference to notions of probabilistic models and information theory during the 1950s and into the 1960s, as one of the quotations above illustrates – the one from Cherry, Halle, and Jakobson 1951. See Hockett 1955, Goldsmith 2001a.

[4] This comes from Jim Breen’s (Monash University) extraordinary resources on Japanese made available at http://www.csse.monash.edu.au/~jwb/wwwjdic.html, and I am extremely indebted to him for making this work easily accessible to the research community. Breen notes that he uses “wa-puro-” Hepburn romaji; thus “long vowels in gairaigo are represented with a ‘-’; long vowels in Japanese words are written with the usual Japanese vowel, which is usually a ‘u’, and sometimes ‘o’ or ‘i’….the voiced ‘tsu’ syllable is written ‘dzu’, not ‘zu’. Similarity the voiced ‘chi’ is written ‘dji’. This is to distinguish them from the voiced ‘su’ and ‘shi’ syllables,

which are written as ‘zu’ and ‘ji’.” I have made the following changes in the phonological representations: sh is changed to S; ssh to SS; ch to C; cch to CC; tsu to tu, other ts to T, and ‘-’ to ‘:’.

[5] This method also reveals errors that have crept into the dictionaries by the dictionary makers.

[6] To be a bit clearer, I am not saying that one *cannot* (or should not) do psycholinguistics, that is, the analysis of how language is used by people. One can, and the tools for doing this are getting better all the time; indeed, these tools have virtually revolutionized the field in the last decade. My point is rather that traditional linguistics, which studies sentences and corpora, has a scientific grounding that is distinct from that of psycholinguistics. This perspective is quite different – indeed, at odds with – both the view (which I find curious) that linguistics is a branch of the biosciences, as Chomsky is wont to say (e.g., Chomsky 1999), or that linguistics studies objects with an ontological status much like that of mathematical objects; Katz and Postal in a number of publications over the last twenty years have discussed a conception of linguistics in this vein.

[7] The discussion in the text overlooks the importance of the complexity of the theory (or model) being used to understand the data. The greater the complexity of the theory, the less explanation is being provided, and one means to making such a statement quantitatively explicit is provided by Minim mum Description Length; see Rissanen 1989, for example. The presentation of this paper in August 2001 was linked to a following paper on the use of MDL in the automatic learning of morphology. The central notion to MDL is that there is a trade-off between grammar complexity and the degree of explanation that a grammar provides to a set of observations, and in particular that the correct grammar has been found when the marginal increase in log probability of a corpus equals the marginal increase in optimal length of the grammar, both expressed in bits of information. What is surprising is that this simple formulation can be made concrete and calculable.

[8] The reader who truly understands the nature of a distribution may cringe at that statement, as I very nearly do; but if the statement is technically objectionable, it is pedagogically reasonable.

[9] Confusingly, this is often called the *negative log probability!* -- because it is the log probability multiplied by negative 1.

[10] Log probability falls roughly into the set of those properties that a physicist might call an *extensive* property: if we divide a larger object up into two smaller pieces, an extensive property is one (like mass) for which the property of the whole is equal to the sum of the properties of the parts (unlike, say, temperature).

[11] We could alternatively consider the possibility that it is dependent on the phoneme that follows it; it turns out that this is mathematically identical.

[12] It is composed essentially of k^2 numbers, where k is the number of distinct phonemes in the language.

[13] I have run experiments that are slightly more complex than what is indicated in the text, forcing a choice between 5 European languages, and getting 98%+ correct results after five words, with natural text.

[14] Especially as described in Kiparsky 1982.

[15] It is important to remember that positivism has been a liberating and a revolutionary philosophy at other times in the past. August Comte, in his *A General View of Positivism* (1856), writes, "Our doctrine, therefore, is one which renders hypocrisy and oppression alike impossible. And it now stands forward as the result of all the efforts of the past, for the regeneration of order, which, whether considered individually or socially, is so deeply compromised by the anarchy of the present time. It establishes a fundamental principle by which true philosophy and sound polity are brought into correlation; a principle which can be felt as well as proved, and which is at once the keystone of a system and a basis of government. I shall show, moreover, in the fifth chapter, that the doctrine is as rich in aesthetic beauty as in philosophical power and in social influence. This will complete the proof of its efficacy as the centre of a universal system. Viewed from the moral, scientific, or poetical aspect, it is equally valuable; and it is the only principle which can bring Humanity safely through the most formidable crisis that she has ever yet undergone." Ernst Mach was the reigning positivist at the end of the 19th century among philosophers of science, and it has often been remarked (among others, by Einstein himself) that it was Mach's remorseless positivism which enabled creative spirits, like Einstein, to question and eventually to overthrow the Newtonian conception of time and space. Closer to home, the great Bantuist A.E. Meeussen was able to imagine, and publish, the finest tonological analyses of his day because he felt no need to justify his analyses beyond their ability to organize complex data.

References

Charniak, Eugene (1993). *Statistical Language Learning*. Cambridge MA: MIT Press.

Cherry, Colin., Morris Halle, et al. (1953). "Toward the logical description of languages in their phonemic aspect." *Language* 29: 34-46.

Chomsky, Noam (1999) "Derivation by Phase'', Cambridge, Massachusetts, *MIT Occasional Papers in Linguistics Number 18*, Distributed by MIT Working Papers in Linguistics**.**

Coleman, John. S. and Janet Pierrehumbert.(1997). Stochastic phonological grammars and acceptability. Third Meeting of the ACL Special Interest Group in Computational Phonology. Somerset NJ,: Association for Computational Linguistics.

Comte, Auguste (1856). *A General View of Positivism*. [Robert Speller & Sons. 1957].

Goldsmith, John (1990). *Autosegmental and Metrical Phonology*. Oxford and Cambridge MA: Basil Blackwell .

Goldsmith, John (1993). Harmonic phonology. in _*The Last Phonological Rule: Reflections on Constraints and*

Derivations. John Goldsmith. Chicago: University of Chicago Press**:** 21--60. [Circulated in 1989]

Goldsmith, John (1995). Introduction. *Handbook of Phonological Theory*. J. A. Goldsmith. Oxford: Basil Blackwell.

Goldsmith, John (2001a). “On information theory, entropy and phonology in the 20th century.” *Folia Linguistica* XXXIV(1-2): 85-100.

Goldsmith, John (2001b). “Unsupervised Learning of the Morphology of a Natural Language.” *Computational Linguistics* 27(2): 153-198.

Hockett, Charles F. (1955). *A manual of phonology*. Baltimore: Waverly Press.

Jurafsky, Daniel and James Martin (2000). *Speech and Language Processing*. Upper Saddle River, NJ: Prentice Hall.

Katz, Jerrold J. (1984) “An Outline of Platonist Grammar”, in Thomas G. Bever, John M. Carroll and Lance A. Miller (eds.) *Talking Minds. The Study of Language in Cognitive Science, Cambridge, MA: MIT Press.*

Katz, Jerrold J. and Paul M. Postal (1991) “Realism vs. Conceptualism in Linguistics”, Linguistics and Philosophy 14: 515-554.

Kiparsky, Paul (1982). Lexical phonology and morphology. _*Linguistics in the Morning Calm*. I. S. Yang. Seoul: Hanshin. 2**:** 3--91.

Rissanen, Jorma (1989). *Stochastic complexity in statistical inquiry*. Singapore ; Teaneck, NJ: World Scientific.

Solomonoff, Ray (1977 [1995]). The discovery of algorithmic probability: a guide for the programming of true creativity. *Journal of Computer and System Sciences*, Vol. 55, No. 1.

This paper appeared in: *Phonological Studies* #5: 21-46.