

**Probability for linguists**  
**John Goldsmith**  
**April 2001**

## **1. Introduction**

---

Probability is playing an increasingly large role in computational linguistics and machine learning, and will be of great importance to us. If you've had any exposure to probability at all, you're likely to think of cases like rolling dice. If you roll one die, there's a 1 in 6 chance -- about 0.166 -- of rolling a "1", and likewise for the five other normal outcomes of rolling a die. Games of chance, like rolling dice and tossing coins, are important illustrative cases in most introductory presentations of what probability is about. This is only natural; the study of probability arose through the analysis of games of chance, only becoming a bit more respectable when it was used to form the rational basis for the insurance industry. But neither of these applications lends itself to questions of linguistics, and linguists tend to be put off by examples like these, examples which seem to suggest that we take it for granted that the utterance of a word is a bit like the roll of a die -- which it's not, as we perfectly well know.

Probability is better thought of in another way. We use probability theory in order to talk in an explicit and quantitative way about the degree of certainty, or uncertainty, that we possess about a question. Putting it slightly differently, if we wanted to develop a theory of how certain a perfectly rational person could be of a conclusion in the light of specific data, we'd end up with something very much like probability theory. And that's how we should think of it.

Let's take an example. Many of the linguistic examples we consider will be along the lines of what a speech recognition system must deal with, which is to say, the task of deciding (or guessing) what word has just been uttered, given knowledge of what the preceding string of words has been coming out of the speaker's mouth. Would you be willing to consider the following suggestions?

Let us suppose that we have established that the person is speaking English. Can we draw any conclusions independent of the sounds that the person is uttering at this moment? Surely we can. We can make an estimate of the probability that the word is in our desk-top Webster's Dictionary, and we can make an estimate of the probability that the word is "the", and an estimate of the probability that the word is -- let's choose another word -- "telephone". We can be quite certain, in fact, that "the" is the most likely word to be produced by an English speaker; as much as five percent of a speaker's words may be "the"s.

## **2. Let's take a look at -- or review -- some of the very basics.**

---

We're going to try to look at language from the roll-of-the-die point of view for a little while. It's not great, but it might just be the best way to start.

The very first notion to be familiar with is that of a *distribution*: a set of (non-negative) numbers that add up to 1.0. In every discussion of probability, distributions play a central role, and one must always ask oneself what is being treated as forming a distribution. Probabilities are always members of a distribution.

Let's consider the roll of a die. There are six results of such a roll, and we typically assume that their probabilities must be equal; it follows that their probabilities must be  $1/6$ , since they add up to 1.0: they form a distribution. We call a distribution in which all values are the same a *uniform distribution*. We always assume that there is a *universe of basic outcomes*, and each outcome has associated with it a probability. The universe of basic outcomes is normally called the *sample space*. The sum of the probabilities of all of the outcomes is 1.0 Any *set* of the outcomes has a probability, which is the sum of the probabilities of the members of the subset. Thus the probability of rolling an even number is 0.5.

In this simplest case, we took the universe of outcomes to consist of 6 members, the numbers 1 through 6. But this is not necessary. We can take the universe of outcomes to be all possible outcomes of two successive rolls of a die. The universe then has 36 members, and the outcome "The first roll is a 1" is not a single member of the universe of outcomes, but rather it is a subset consisting of 6 different members, each with a probability of  $1/36$ . These six are: (1) The first roll is 1 and the second is 1; (2) The first roll is 1 and the second is 2; ... (6) The first roll is 1 and the second is 6. The probability of these 6 add up to  $1/6$ .

It is not hard to see that if a universe consists of  $N$  rolls of a die ( $N$  can be any positive number), the number of outcomes in that universe will be  $6^N$ . (And the probability of any particular sequence is  $1/6^N$ , if the distribution is uniform).

Be clear on the fact that whenever we pose a question about probability, we have to specify precisely what the universe of outcomes (i.e., the sample space) is that we're considering. It matters whether we are talking about the universe of *all* possible sequences of 6 rolls of a die, or all possible sequences of 6 or fewer rolls of a die, for example. You should convince yourself that the latter universe is quite a bit bigger, and hence the probability of any die-roll that is 6 rolls long will have a *lower* probability in that *larger* universe than it does in the universe consisting only of 6 rolls of a die.

We have just completed our introduction to the most important ideas regarding probabilistic models. Never lose sight of this: we will be constructing a model of a set of data and we will assign a distribution to the basic events of that universe. We will almost certainly assign that distribution via some simpler distributions assigned to a simpler universe. For example, the complex universe may be the universe of all ways of rolling a die 6 or fewer times, and the simpler universe will be single rolls of a fair, six-sided die. From the simple, uniform distribution on single rolls of a die we will build up a distribution on a more complex universe.

Notation, or a bit more than notation: It should always be possible to write an equation summing probabilities over the distribution so they add up to 1.0:

$$\sum_i p_i = 1.0.$$

You should be able to write this for any problem that you tackle.

We can imagine the universe to consist of a sequence of rolls of a die anywhere in length from 1 roll to (let us say) 100. The counting is a little more complicated, but it's not all that different. And each one of them is equally likely (and not very likely, as you can convince yourself).

Let's make the die bigger. Let us suppose, now, that we have a large die with 1,000 sides on it. We choose the 1,000 most frequent words in a large corpus -- say, the Brown corpus. Each time we roll the die, we choose the word with the corresponding rank, and utter it. That means that each time the die comes up "1" (which is only once in a thousand rolls, on average), we say the word "the". When it comes up "2", we say "of" -- these are the two most frequent words. And so forth.

If we start rolling the die, we'll end up with utterances like the following:

320 990 646 94 756

which translates into: *whether designed passed must southern.*

That's what this worst of random word generators would generate. But that's not what we're thinking about grammars probabilistically to do -- not at all. Rather, what we're interested in is the probability that this model would assign to a particular sentence that somebody has already uttered. Let's use, as our example, the sentence: ***In the beginning was the word.***

There are six words in this sentence, and it just so happens that all six are among the 1,000 most common words in the Brown corpus. So the probability that we would *assign* to this sentence is  $1/1000 * 1/1000 * 1/1000 * 1/1000 * 1/1000 * 1/1000$ , which can also be expressed more readably as  $10^{-18}$ . There are  $1,000 = 10^3$  events in the universe of strings of one word in length, and  $1,000,000 = 10^6$  events in the universe of strings of 2 words in length, and  $10^{18}$  events in the universe of strings of 6 words. That is why each such event has a probability of the reciprocal of that number. (If there are K events which are equally likely, then each has the probability  $1/K$ , right?)

I hope it is already clear that this model would assign that probability to *any* sequence of six words (if the words are among the lexicon that we possess). Is this good or bad? It's neither the one nor the other. We might say that this is a terrible grammar of English, but such a judgment might be premature. This method will assign a zero probability to any sequence of words in which at least one word does not appear in the top 1000 words of the Brown corpus. That may sound bad, too, but do notice that it means that such a grammar will assign a zero probability to any sentence in a language that is not English. And it will assign a non-zero probability to any word-sequence made up entirely of words from the top 1,000 words.

We could redo this case and include a non-zero probability for *all* of the 47,885 distinct words in the Brown Corpus. Then any string of words all of which appear in the corpus will be assigned a probability of  $(1/47,885)^N$ , where N is the number of words in the string, assuming a sample space of sentences all of length N. A sentence of 6 words would be assigned a probability of  $(1/47,885)^6$ , which just so happens to be about  $(2.1 * 10^{-5})^6$ , or **86 \*  $10^{-30}$** . We'll get back to that (very small) number in a few paragraphs.

Or -- we could do better than that (and the whole point of this discussion is so that I can explain in just a moment exactly what "doing better" really means in this context). We could assign to each word in the corpus a probability equal to its frequency in the corpus. The word "the", for example, appears 69,903 out of the total 1,159,267 words, so its probability will be approximately .0603 -- and other words have a much lower probability. "leaders" occurs 107 times, and thus would be assigned the probability  $107/1,159,267 = .000092$  (it is

the 1,000th most frequent word). Is it clear that the sum of the probabilities assigned to all of the words adds up to 1.00? It should be.

This is very important, and most of what we do from now on will assume complete familiarity with what we have just done, which is this: we have a universe of outcomes, which are our words, discovered empirically (we just took the words that we encountered in the corpus), and we have assigned a probability to them which is exactly the frequency with which we encountered them in the corpus. We will call this a *unigram model* (or a *unigram word model*, to distinguish it from the parallel case where we treat letters or phonemes as the basic units). The probabilities assigned to each of the words adds up to 1.0

Table 1: Top of the unigram distribution for the Brown Corpus.

word	count	frequency
the	69903	0.068271
of	36341	0.035493
and	28772	0.028100
to	26113	0.025503
a	23309	0.022765
in	21304	0.020807
that	10780	0.010528
is	10100	0.009864
was	9814	0.009585
he	9799	0.009570
for	9472	0.009251
it	9082	0.008870
with	7277	0.007107
as	7244	0.007075
his	6992	0.006829
on	6732	0.006575
be	6368	0.006219
s	5958	0.005819
i	5909	0.005771
at	5368	0.005243

(Note that "s" is the possessive s, being treated as a distinct word.)

Now let's ask what the probability is of the sentence "the woman arrived." To find the answer, we must, first of all, specify that we are asking this question in the context of sentence composed of 3 words -- that is, sentence of length 3. Second, in light of the previous paragraph, we need to find the probability of each of those words in the Brown Corpus. The probability of "the" is 0.068 271; prob (woman) = 0.000 23; prob (arrived) = .000 06. These numbers represent their probabilities where the universe in question is a universe of single words being chosen from the universe of possibilities -- their probabilities in a unigram word model. What we are interested in now is the universe of 3-word sentences. (By the way, I am using the word "sentence" to mean "sequence of words" -- use of that term doesn't imply a claim about grammaticality or acceptability.) We need to be able to talk about sentences whose first word is "the", or whose second word is "woman";

let's use the following notation. We'll indicate the word number in square brackets, so if  $S$  is the sentence "the woman arrived," then  $S[1] = \text{"the"}$ ,  $S[2] = \text{"woman"}$ , and  $S[3] = \text{"arrived"}$ . We may also want to refer to words in a more abstract way -- to speak of the  $i^{\text{th}}$  word, for example. If we want to say the first word of sentence  $S$  is the  $i^{\text{th}}$  word of the vocabulary, we'll write  $S[1] = w_i$ . (This is to avoid the notation that Charniak and others use, which I think is confusing, and which employs both subscripts and superscripts on  $w$ 's.)

We need to assign a probability to each and every sequence (i.e., sentence) of three words from the Brown Corpus in such a fashion that these probabilities add up to 1.0. The natural way to do that is to say that the probability of a sentence is the product of the probabilities: if  $S = \text{"the woman arrived"}$  then

$$(1) \quad \text{prob}(S) = \text{prob}(S[1] = \text{"the"}) * \text{prob}(S[2] = \text{"woman"}) * \text{prob}(S[3] = \text{"arrived"})$$

and we do as I suggested, which is to suppose that the probability of a word is independent of what position it is in. We would state that formally:

For all sentences  $S$ , all words  $w$  and all positions  $i$  and  $j$ :

$$\text{prob}(S[i] = w_n) = \text{prob}(S[j] = w_n).$$

A model with that assumption is said to be a *stationary* model. Be sure you know what this means. For a linguistic model, it seems reasonable, but it isn't just a logical truth. In fact, upon reflection, you will surely be able to convince yourself that the probability of the *first* word of a sentence being "the" is vastly greater than the probability of the *last* word in the sentence being "the". Thus a stationary model is not the last word (so to speak) in models.

Sometimes we may be a bit sloppy, and instead of writing " $\text{prob}(S[i] = w_n)$ " (which in English would be "the probability that the  $i^{\text{th}}$  word of the sentence is word number  $n$ ") we may write " $\text{prob}(S[i])$ ", which in English would be "the probability of the  $i^{\text{th}}$  word of the sentence". You should be clear that it's the first way of speaking which is proper, but the second way is too easy to ignore.

You should convince yourself that with these assumptions, the probabilities of all 3-word sentences does indeed add up to 1.0.

Exercise 1. Show mathematically that this is correct.

As I just said, the natural way to assign probabilities to the sentences in our universe is as in (1); we'll make the assumption that the probability of a given word is stationary, and furthermore that it is its empirical frequency (i.e., the frequency we observed) in the Brown Corpus. So the probability of "the woman arrived" is  $0.068\,271 * 0.000\,23 * .00006 = 0.000\,000\,000\,942\,139\,8$ , or about  $9.42 * 10^{-10}$ .

What about the probability of the sentence "in the beginning was the word"? We calculated it above to be  $10^{-18}$  in the universe consisting of all sentences of length 6 (exactly) where the words were just the 1,000 most frequency words in the Brown Corpus, with uniform

distribution. And the probability was  $8.6 * 10^{-29}$  when we considered the universe of all possible sentences of six words in length, where the size of the vocabulary was the whole vocabulary of the Brown Corpus, again with uniform distribution. But we have a new *model* for that universe, which is to say, we are considering a different distribution of probability mass. In the new model, the probability of the sentence is the product of the empirical frequencies of the words in the Brown Corpus, so the probability of *in the beginning was the word* in our new model is

$$\begin{aligned} &.021 \quad * \quad .068 \quad * \quad .00016 \quad * \quad .0096 \quad * \quad .021 \quad * \quad .00027 = \\ &2.1 \quad * 10^{-2} * 6.8 \quad * 10^{-2} * 1.6 * 10^{-4} * 9.6 * 10^{-3} * 2.1 * 10^{-2} * 2.7 * 10^{-4} = \\ &1243 * 10^{-17} = \\ &1.243 * 10^{-14}. \end{aligned}$$

That's a much larger number than we got with other distributions (remember, the exponent here is -14, so this number is *greater* than one which has a more negative exponent.)

The main point for the reader now is to be clear on what the significance of these two numbers is:  $10^{-18}$  for the first model,  $8.6 * 10^{-29}$  for the second model, and  $1.243 * 10^{-14}$  for the third. But it's the same sentence, you may say! Why the different probabilities? The difference is that a higher probability (a bigger number, with a smaller negative exponent, putting it crudely) is assigned to the sentence that we know is an English sentence in the frequency-based model. If this result holds up over a range of real English sentences, this tells us that the frequency-based model is a better model of English than the model in which all words have the same frequency (a uniform distribution). That speaks well for the frequency-based model. In short, we prefer a model that scores better (by assigning a higher probability) to sentences that *actually and already exist* -- we prefer that model to any other model that assigns a lower probability to the actual corpus.

In order for a model to assign higher probability to actual and existing sentences, it must assign less probability to other sentences (since the total amount of probability mass that it has at its disposal to assign totals up to 1.000, and no more). So of course it assigns lower probability to a lot of unobserved strings. On the frequency-based model, a string of word-salad like *civilized streams riverside prompt shaken squarely* will have a probability even lower than it does in the uniform distribution model. Since each of these words has probability  $1.07 * 10^{-5}$  (I picked them that way --), the probability of the sentence is  $(1.07 * 10^{-5})^6 = 1.4 * 10^{-30}$ . That's the probability based on using empirical frequencies. Remember that a few paragraphs above we calculated the probability of any six-word sentence in the uniform-distribution model as  $8.6 * 10^{-29}$ ; so we've just seen that the frequency-based model gives an even *smaller* probability to this word-salad sentence than did the uniform distribution model -- which is a good thing.

You are probably aware that so far, our model treats word order as irrelevant -- it assigns the same probability to *beginning was the the in word* as it does to *in the beginning was the word*. We'll get to this point eventually.

## Probability mass

---

It is sometimes helpful to think of a distribution as a way of sharing an abstract goo called *probability mass* around all of the members of the universe of basic outcomes (that is, the sample space). Think of there being 1 kilogram of goo, and it is cut up and assigned to the various members of the universe. None can have more than 1.0 kg, and none can have a negative amount, and the total amount must add up to 1.0 kg. And we can modify the model by moving probability mass from one outcome to another if we so choose.

### **Conditional probability**

---

I stressed before that we must start an analysis with some understanding as to what the universe of outcomes is that we are assuming. That universe forms the background, the given, of the discussion. Sometimes we want to shift the universe of discussion to a more restricted sub-universe – this is *always* a case of having additional information, or at least of acting as if we had additional information. This is the idea that lies behind the term *conditional probability*. We look at our universe of outcomes, with its probability mass spread out over the set of outcomes, and we say, let us consider only a sub-universe, and ignore all possibilities outside of that sub-universe. We then must ask: how do we have to change the probabilities inside that sub-universe so as to ensure that the probabilities inside it add up to 1.0 (to make it a distribution)? Some thought will convince you that what must be done is to divide the probability of each event by the total amount of probability mass inside the sub-universe.

There are several ways in which the new information which we use for our conditional probabilities may come to us. If we are drawing cards, we may somehow get new but incomplete information about the card -- we might learn that the card was red, for example. In a linguistic case, we might have to guess a word, and then we might learn that the word was a noun. A more usual linguistic case is that we have to guess a word when we know what the preceding word was. But it should be clear that all three examples can be treated as similar cases: we have to guess an outcome, but we have some case-particular information that should help us come up with a better answer (or guess).

Let's take another classic probability case. Let the universe of outcomes be the 52 cards of a standard playing card deck. The probability of drawing any particular card is  $1/52$  (that's a uniform distribution). What if we restrict our attention to red cards? It might be the case, for example, that of the card drawn, we know it is red, and that's all we know about it; what is the probability now that it is the Queen of Hearts?

The sub-universe consisting of the red cards has probability mass 0.5, and the Queen of Hearts lies within that sub-universe. So if we restrict our attention to the 26 outcomes that comprise the "red card sub-universe," we see that the sum total of the probability mass is only 0.5 (the sum of 26 red cards, each with  $1/52$  probability). In order to consider the sub-universe as having a distribution on it, we must divide each of the  $1/52$  in it by 0.5, the total probability of the sub-universe in the larger, complete universe. Hence the probability of the Queen of Hearts, *given the Red Card sub-Universe* (given means *with the knowledge that the event that occurs is in that sub-universe*), is  $1/52$  divided by  $1/2$ , or  $1/26$ .

This is traditionally written:  $p(A|B)$  = probability of A, given B =  

$$prob(A | B) = \frac{prob(A \text{ and } B)}{prob(B)} .$$

### **Guessing a word, given knowledge of the previous word:**

---

Let's assume that we have established a probability distribution, the unigram distribution, which gives us the best estimate for the probability of a randomly chosen word. We have done that by actually measuring the frequency of each word in some corpus. We would like to have a better, more accurate distribution for estimating the probability of a word, conditioned by knowledge of what the preceding word was. There will be as many such distributions as there are words in the corpus (one less, if the last word in the corpus only occurs there and nowhere else.) This distribution will consist of these probabilities:

$p_k( S[i] = w_j \text{ given that } S[i-1] = w_k )$ , which is usually written in this way:

$p_k( S[i] = w_j \mid S[i-1] = w_k )$

The probability of "the" in an English corpus is very high, but not if the preceding word is "the" -- or if the preceding word is "a", "his", or lots of other words.

I hope it is reasonably clear to you that so far, (almost) nothing about language or about English in particular has crept in. The fact that we have considered conditioning our probabilities of a word based on what word *preceded* is entirely arbitrary; as we see in Table 4, we could just as well look at the conditional probability of words conditioned on what word follows, or even conditioned on what the word was two words to the left. In Table 5, we look at the distribution of words that appear two words to the right of "the". As you see, I treat punctuation (comma, period) as separate words. Before continuing with the text below these tables, look carefully at the results given, and see if they are what you might have expected if you had tried to predict the result ahead of time.

Table 2: Top of the Brown Corpus for words following "the":

Total count 69936

	word	count	count / 69936
0	first	664	0.00949439487531457
1	same	629	0.00899393731411576
2	other	419	0.0059911919469229
3	most	419	0.0059911919469229
4	new	398	0.00569091741020361
5	world	393	0.0056194234728895
6	united	385	0.00550503317318691
7	*j	299	0.00427533745138412



8	state	292	0.00417524593914436
9	two	267	0.00381777625257378
10	only	260	0.00371768474033402
11	time	250	0.00357469686570579
12	way	239	0.00341741020361473
13	old	234	0.00334591626630062
14	last	223	0.00318862960420956
15	house	216	0.0030885380919698
16	man	214	0.00305994051704415
17	next	210	0.00300274536719286
18	end	206	0.00294555021734157
19	fact	194	0.00277396476778769
20	whole	190	0.0027167696179364

Table 3: Top of the Brown Corpus for words following "of".  
Total count 36388

	word	count	count / 36,388
1	the	9724	0.267230955259976
2	a	1473	0.0404803781466418
3	his	810	0.0222600857425525
4	this	553	0.015197317797076
5	their	342	0.00939870286907772
6	course	324	0.008904034297021
7	these	306	0.00840936572496427
8	them	292	0.00802462350225349
9	an	276	0.00758491810486974
10	all	256	0.00703528635814005
11	her	252	0.00692536000879411
12	our	251	0.00689787842145762
13	its	229	0.00629328350005496
14	it	205	0.00563372540397933
15	that	156	0.00428712762449159
16	*j	156	0.00428712762449159
17	such	140	0.00384742222710784
18	those	135	0.00371001429042541
19	my	128	0.00351764317907002
20	which	124	0.00340771682972408

Table 4: Top of the Brown Corpus for words preceding "the".  
Total count 69936

	word	count	count / 69,936
1	of	9724	0.139041409288492
2	.	6201	0.0886667810569664
3	in	6027	0.0861787920384351

4	,	3836	0.0548501487073896
5	to	3485	0.0498312743079387
6	on	2469	0.0353037062457104
7	and	2254	0.0322294669412034
8	for	1850	0.0264527568062228
9	at	1657	0.023693090825898
10	with	1536	0.0219629375428964
11	from	1415	0.0202327842598948
12	that	1397	0.0199754060855639
13	by	1349	0.0192890642873484
14	is	799	0.0114247311827957
15	as	766	0.0109528711965225
16	into	675	0.00965168153740563
17	was	533	0.00762125371768474
18	all	430	0.00614847860901396
19	when	418	0.00597689315946008
20	but	389	0.00556222832303821

Table 5: Top of the Brown Corpus for words 2 to the right of "the".  
Total count 69936

	word	count	count / 69,936
1	of	10861	0.155299130633722
2	.	4578	0.0654598490048044
3	,	4437	0.0634437199725463
4	and	2473	0.0353609013955617
5	to	1188	0.0169869595058339
6	'	1106	0.0158144589338824
7	in	1082	0.0154712880347747
8	is	1049	0.0149994280485015
9	was	950	0.013583848089682
10	that	888	0.012697323266987
11	for	598	0.00855067490276825
12	were	386	0.00551933196064974
13	with	370	0.00529055136124457
14	on	368	0.00526195378631892
15	states	366	0.00523335621139327
16	had	340	0.00486158773735987
17	are	330	0.00471859986273164
18	as	299	0.00427533745138412
19	at	287	0.00410375200183024
20	or	284	0.00406085563944178

What do we see? Look at Table 2, words following "the". One of the most striking things is how few nouns, and how many adjectives, there are among the most frequent words here -- that's probably not what you would have guessed. None of them are very high in frequency; none place as high as 1 percent of the total. In Table 3, however, the words after "of", one word is over 25%: "the". So *not all words are equally helpful in helping to guess what the next word is*. In Table 4, we see words preceding "the", and we notice that other than punctuation, most of these are prepositions. Finally, in Table 5, we see that if you know a word is "the", then the probability that the word-after-next is "of" is greater than 15% -- which is quite a bit.

Exercise 2: What do you think the probability distribution is for the 10<sup>th</sup> word after "the"? What are the two most likely words? Why?

Conditions can come from other directions, too. For example, consider the relationships of English letters to the phonemes they represent. We can ask what the probability of a given phoneme is -- not conditioned by anything else -- or we can ask what the probability of a phoneme is, given that it is related to a specific letter.

### More conditional probability: Bayes' Rule

---

Let us summarize. How do we calculate what the probability is that the n<sup>th</sup> word of a sentence is "the" if the n-1<sup>st</sup> word is "of"? We count the number of occurrences of "the" that follow "of", and divide by the total number of "of"s.

Total number of "of":	36,341
Total number of "of the":	9,724

In short,  $p(S[i] = \text{the} \mid S[i-1] = \text{of}) = 9724 / 36341 = 0.267$

What is the probability that the n<sup>th</sup> word is "of", if the n+1<sup>st</sup> word is "the"? We count the number of occurrences of "of the", and divide by the total number of "the": that is,  $p(S[i] = \text{of} \mid S[i+1] = \text{the}) = 9,724 / 69,903 = 0.139$

This illustrates the relationship between  $p(A \mid B)$  "the probability of A given B" and  $p(B \mid A)$  "the probability of B given A". This relationship is known as Bayes' Rule. In the case we are looking at, we want to know the relationship between the probability of a word being *the*, given that the preceding word was *of* -- and the probability that a word is *of*, given that the next word is *the*.

$p(S[i] = \text{of} \mid S[i+1] = \text{the}) = p(S[i] = \text{of} \text{ and } S[i+1] = \text{the}) / p(S[i+1] = \text{the})$

and also, by the same definition:

$p(S[i] = \text{the} \mid S[i-1] = \text{of}) = p(S[i] = \text{of} \text{ and } S[i+1] = \text{the}) / p(S[i-1] = \text{of})$

Both of the preceding two lines contain the phrase:  $p(S[i] = \text{"of"} \text{ and } S[i+1] = \text{"the"})$ . Let's solve both equations for that quantity, and then equate the two remaining sides.

$$p(S[i] = \text{"of"} | S[i+1] = \text{"the"}) * p(S[i+1] = \text{"the"}) = p(S[i] = \text{"of"} \text{ and } S[i+1] = \text{"the"})$$

$$p(S[i] = \text{"the"} | S[i-1] = \text{"of"}) * p(S[i-1] = \text{"of"}) = p(S[i] = \text{"of"} \text{ and } S[i+1] = \text{"the"})$$

Therefore:

$$p(S[i] = \text{"of"} | S[i+1] = \text{"the"}) * p(S[i+1] = \text{"the"}) = p(S[i] = \text{"the"} | S[i-1] = \text{"of"}) * p(S[i-1] = \text{"of"})$$

And we will divide by " $p(S[i+1] = \text{"the"})$ ", giving us:

$$p(S[i] = \text{"of"} | S[i+1] = \text{"the"}) = \frac{p(S[i] = \text{"the"} | S[i-1] = \text{"of"}) * p(S[i-1] = \text{"of"})}{p(S[i+1] = \text{"the"})}$$

And writing that without the equation editor, which may not make it to HTML :

$$p(S[i] = \text{"of"} | S[i+1] = \text{"the"}) = \frac{p(S[i] = \text{"the"} | S[i-1] = \text{"of"}) * p(S[i-1] = \text{"of"})}{p(S[i+1] = \text{"the"})}$$

The general form of Bayes' Rule is:

$$\text{prob}(A | B) = \frac{p(A)}{p(B)}$$

$$\text{again, that's: } \text{prob}(A | B) = p(B | A) p(A) / p(B)$$

## The joy of logarithms

It is, finally, time to get to logarithms -- I heave a sigh of relief. Things are much simpler when we can use logs. Let's see why.

In everything linguistic that we have looked at, when we need to compute the probability of a string of words (or letters, etc.), we have to multiply a string of numbers, and each of the numbers is quite small, so the product gets extremely small very fast. In order to avoid such small numbers (which are hard to deal with in a computer), we will stop talking about probabilities, much of the time, and talk instead about the *logarithms* of the probabilities -- or rather, since the logarithm of a probability is always a negative number and we prefer positive numbers, we will talk about -1 times the log of the probability. Let's call that the *positive log probability*. If the probability is  $p$ , then we'll write the positive log probability as  $\{p\}$ .

Notation: if  $p$  is a number:  $\{p\} = -1 * \log p$   
 If  $E$  is an event, then  $\{E\} = -1 * \log \text{prob}(E)$

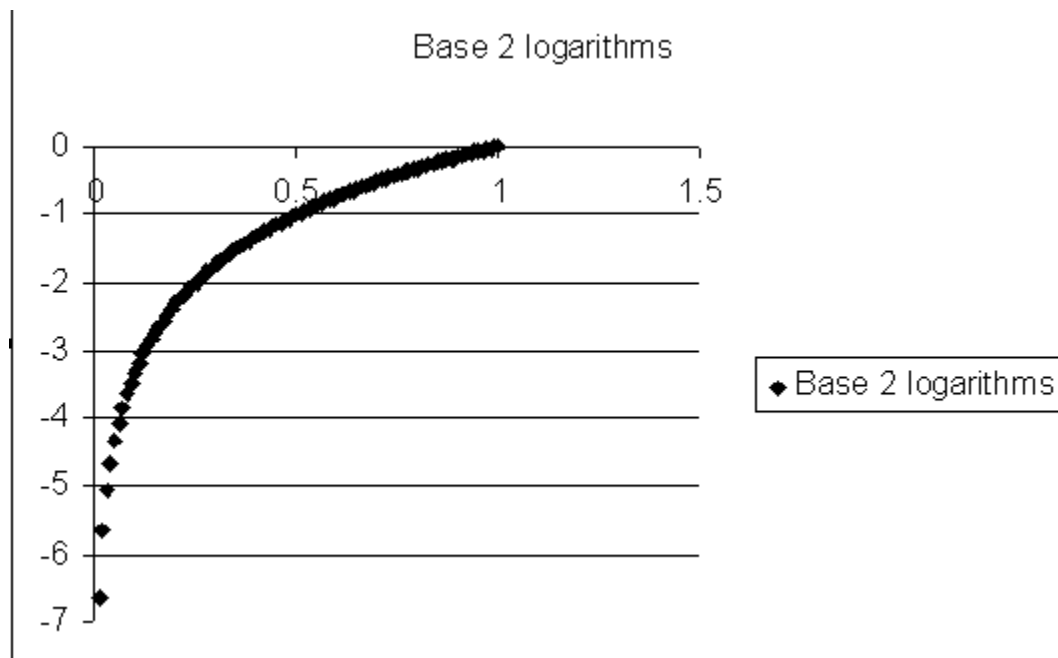
(One standard notation puts a tilde over the p, but it's hard to put a tilde over a long formula.) As a probability gets very small, its positive log probability gets larger, but at a much, much slower rate, because when you multiply probabilities, you just add positive log probabilities. That is,

$$\log ( \text{pr}(S[1]) * \text{pr}( S[2] ) * \text{pr}( S[3] ) * \text{pr}( S[4] ) ) =$$

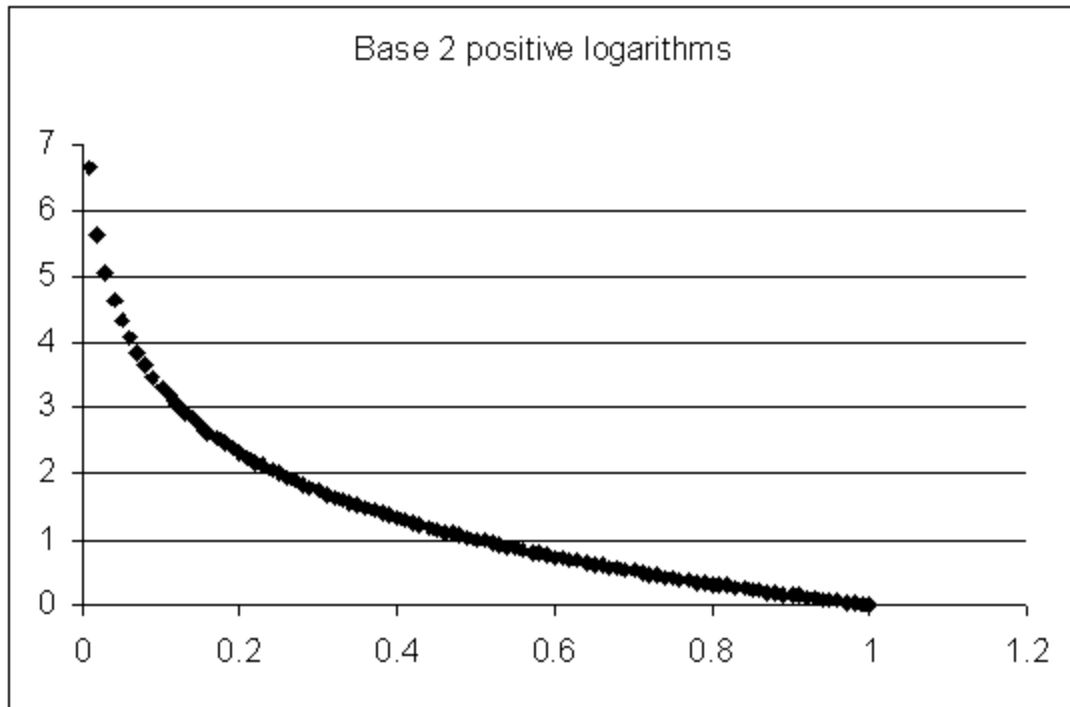
$$-1 * \{ S[1] \} + \{ S[2] \} + \{ S[3] \} + \{ S[4] \}$$

And then it becomes possible for us to do such natural things as inquiring about the average log probability -- but we'll get to that.

At first, we will care about the logarithm function for values in between 0 and 1, which is where all probabilities necessarily lie, as in the graph below:



If we make these positive log probabilities, we get the following graph:



It's important to be comfortable with notation, so that you see easily that the preceding equation can be written as follows, where the left side uses the capital pi to indicate products, and the right side uses a capital sigma to indicate sums:

$$\log \left[ \prod_{i=1}^4 \text{prob}(S[i]) \right] = \sum_{i=1}^4 \log \text{prob}(S[i])$$

We will usually be using base 2 logarithms. You recall that the log of a number  $x$  is the power to which you have to raise the base to get the number  $x$ . If our logs are all base 2, then the log of 2 is 1, since you have to raise 2 to the power 1 to get 2, and log of 8 is 3, since you have to raise 2 to the 3<sup>rd</sup> power in order to get 8 (you remember that 2 cubed is 8). So for almost the same reason, the log of 1/8 is -3, and the positive log of 1/8 is therefore 3.

If we had been using base 10 logs, the logs we'd get would be smaller by a factor of about 3. The base 2 log of 1,000 is almost 10 (remember that 2 to the 10<sup>th</sup> power is 1,024), while the base 10 log of 1,000 is exactly 3.

It almost never makes a difference what base log we use, actually, until we get to information theory. But we will stick to base 2 logs anyway.

**Exercise 3:** Express Bayes' Rule in relation to log probabilities.

**Interesting digression:** There is natural relationship between the real numbers (both positive, negative, and 0) along with the operation of addition, on the one hand, and the positive real numbers along with operation of multiplication:

Real numbers + addition  $\Leftrightarrow$  Positive reals + multiplication

We call such combinations of a set and an operation ("real numbers + addition") *groups*. Zero has a special property with respect to addition: it is the identity element, because one can add zero and make no change; 1 has the same special property (of being the identity element) with respect to multiplication.

So there's this natural relationship between two *groups*, and the natural relationship maps the identity element in the one group to the identity element in the other -- and the relationship preserves the operations. This "natural relationship" maps any element  $x$  in the "Positive reals + multiplication" group to  $\log x$  in the "reals + addition" group, and its inverse operation, mapping from the multiplication group to the addition group is the exponential operation,  $2^x$ . So:  $a \cdot b = \exp ( \log (a) + \log (b) )$ .

$(\text{reals}, +)$   
 $\uparrow \log \quad \downarrow \exp$   
 $(\text{positive reals}, \bullet)$

And similarly, and less interestingly:  $a + b = \log ( \exp (a) \exp (b) )$ .

Exercise 4: Explain in your own words what the relationship is between logarithms and exponentiation (exponentiation is raising a number to a given power).

### Adding log probabilities in the unigram model

The probability of a sentence  $S$  in the unigram model is the product of the probabilities of its words, so the log probability of a sentence in the unigram model is the sum of the log probabilities of its words. That makes it particularly clear that the longer the sentence gets, the larger its log probability gets. In a sense that is reasonable -- the longer the sentence, the less likely it is. But we might also be interested in the *average* log probability of the sentence, which is just the total log probability of the sentence divided by the number of words; or to put it another way, it's the average log probability *per word*  $= \frac{1}{N} \sum_{i=1}^N \{ S[i] \}$ .

This quantity, which will become more and more important as we proceed, is also called the *entropy* -- especially if we're talking about averaging over not just one sentence, but a large, representative sample, so that we can say it's (approximately) the entropy of the language, not just of some particular sentence.

We'll return to the entropy formula, with its initial  $1/N$  to give us an average, but let's stick to the formula that simply sums up the log probabilities  $\sum_{i=1}^N \{ S[i] \}$ . Observe carefully that this is a sum in which we sum over the successive words of the sentence. When  $i$  is 1, we are considering the first word, which might be "the", and when  $i$  is 10, the tenth word might be "the" as well.

In general, we may be especially interested in very long corpora, because it is these corpora which are our approximation to the whole (nonfinite) language. And in such cases, there will be many words that appear quite frequently, of course. It makes sense to re-order the summing of the log probabilities -- because the sum is the same regardless of the order in which you add numbers -- so that all the identical words are together. This means that we can rewrite the sum of the log probabilities as a sum over words in the *vocabulary* (or the dictionary -- a list where each distinct word occurs only once), and multiply the log probability by the number of times it is present in the entire sum. Thus (remember the braces mark positive logs):

$$(\text{sum over words in string}) \quad \sum_{i=1}^N \{ S[i] \} = \sum_{j=1}^V \text{count}(\text{word}_j) \{ \text{word}_j \} \quad (\text{sum over vocabulary})$$

If we've kept track all along of how many words there are in this corpus (calling this "N"), then if we divide this calculation by N, we get, on the left, the average log probability, and,

on the right:  $\sum_{j=1}^V \frac{\text{count}(\text{word}_j)}{N} \{ \text{word}_j \}$ . That can be conceptually simplified some more,

because  $\frac{\text{count}(\text{word}_j)}{N}$  is the proportional frequency with which  $\text{word}_j$  appears in the list of words, which we have been using as our estimate for a word's probability. Therefore we can replace  $\frac{\text{count}(\text{word}_j)}{N}$  by  $\text{freq}(\text{word}_j)$ , and end up with the formula:

$$\sum_{j=1}^V \text{prob}(\text{word}_j) \{ \text{word}_j \},$$

which can also be written as

$$-\sum_{j=1}^V \text{prob}(\text{word}_j) \log \text{prob}(\text{word}_j).$$

This last formula is the formula for the **entropy** of a set, and we will return to it over and over. We can summarize what we have just seen by saying, again, that the entropy of a language is the average log probability of the words.

### Let's compute the probability of a string

---

Let's express the count of a letter  $p$  in a corpus with the notation  $[p]$ , and we'll also allow ourselves to index over the letter of the alphabet by writing " $l_i$ " -- that represents the  $i^{\text{th}}$  letter. Suppose we have a string  $S_1$  of length  $N$ . What is its probability? If we assume that the probability of each letter is independent of its context, and we use its frequency as its probability, then the answer is simply:

$$\prod_{i=1}^N \left( \frac{[l_i]}{N} \right)^{[l_i]}$$



Suppose we add 10  $e$ 's to the end of string  $S_1$ . How does the probability of the new string  $S_2$  compare to  $S_1$ ?

The probability of  $S_2$  is: 
$$\prod_{i=1}^{26} \left( \frac{[l_i]}{N+10} \right)^{[l_i]} \left( \frac{[e]+10}{N+10} \right)^{[e]+10}$$

Let's take the ratio of the two:

$$\frac{\prod_{i=1}^{26} \left( \frac{[l_i]}{N_1} \right)^{[l_i]}}{\prod_{i=1, i \neq e}^{26} \left( \frac{[l_i]}{N_2} \right)^{[l_i]} \left( \frac{[e]+10}{N_2} \right)^{[e]+10}}$$

$$\frac{p_1(e)^{[e]} N_1^{([e]-N_1)} \prod_{i=1, i \neq e}^{26} [l_i]^{[l_i]}}{p_2(e)^{[e]+10} N_2^{([e]+10-N_2)} \prod_{i=1, i \neq e}^{26} [l_i]^{[l_i]}}$$

but  $N_2 = N_1 + 10$ , so this equals

$$\left( \frac{p_1(e)}{p_2(e)} \right)^{[e]} \left( p_2(e) \right)^{-10} \left( \frac{N_1}{N_2} \right)^{[e]-N_1}$$

taking logs:  $[t]_1 \Delta(e) - 10 \log p_2(e) - (N_1 - [e]) \Delta(N)$

where the capital delta function is the log ratio of the values in the before and the after condition (state 1 in the numerator, state 2 in the denominator).

Or in words:

the difference of the log probabilities is the sum of three terms, each weighted by the size of the parts of the string, which are: the original  $e$ 's; 10 new  $e$ 's; and everything else. The first is weighted by the Delta function; the second by the information content of the new  $e$ 's; and the last by a value of approximate  $[e]$  *bits*! (Remember that  $\log(1+x)$  is approximately  $x$ , for small values of  $x$ , because the first derivative of  $\log$  is 1 at 1.)

### Maximizing probability of a sentence, or a corpus

We will now encounter a new and very different idea, but one which is of capital importance: **the fundamental goal of analysis is to maximize the probability of the observed data.** All empirical learning centers around that maxim. Data is important, and learning is possible, because of that principle.

When we have a simple model in mind, applying this maxim is simple; when the models we consider grow larger and more complex, it is more difficult to apply the maxim.

If we restrict ourselves at first to the unigram model, then it is not difficult to prove – but it is important to recognize – that the maximum probability that can be obtained for a given corpus is the one whose word-probabilities coincide precisely with the observed frequencies. It is not easy at first to see what the point is of that statement, but it is important to do so.

Let us remind ourselves that we can assign a probability to a corpus (which is, after all, a specific set of words) with *any* distribution, that is, any set of probabilities that add up to 1.0. If there are words in the corpus which do not get a positive value in the distribution, then the corpus will receive a total probability of zero (remind yourself why this is so!), but that is not an impossible situation. (Mathematicians, by the way, refer to the set which gets a non-zero probability as the *support* of the distribution. Computational linguists may say that they are concerned with making sure that all words are in the support of their probability distribution.)

Suppose we built a distribution for the words of a corpus randomly -- ensuring only that the probabilities add up to 1.0. (Let's not worry about what "randomly" means here in too technical a way.) To make this slightly more concrete, let's say that these probabilities form the distribution  $Q$ , composed of a set of values  $q(\text{word}_i)$ , for each word in the corpus (and possibly other words as well). Even this randomly assigned distribution would (mathematically) assign a probability to the corpus. It is important to see that the probability is equal to

$$(\text{sum over words in string}) \prod_{i=1}^N q(S[i])$$

and that is equal to

$$= \prod_{j=1}^V q(\text{word}_j)^{\text{count}(\text{word}_j)} \quad (\text{sum over vocabulary}).$$

Make sure you understand why this exponent is here: when we multiply together  $k$  copies of the probability of a word (because that word appears  $k$  times in a corpus), the probability of the entire corpus includes,  $k$  times, the probability of that word in the product which is its probability.

If we now switch to thinking about the log probability, any particular word which occurs  $k$  times in the corpus will contribute  $k$  times its log probability to the entire sum which gives us the (positive) log probability:

$$\sum_{j=1}^V \text{count}(\text{word}_j) \log q(\text{word}_j) \quad (\text{sum over vocabulary})$$

What should be clear by now is that we can use *any* distribution to assign a probability to a corpus. We could even use the uniform distribution, which assigns the same probability to each word.

Now we can better understand the idea that we *may* use a distribution for a given corpus whose probabilities are defined exactly by the frequencies of the words in a given corpus. It is a mathematical fact that this "empirical distribution" assigns the *highest* probability to the corpus, and this turns out to be an extremely important property. (Important: you should convince yourself now that if this is true, then the empirical distribution also assigns the lowest entropy to the corpus.)

Exercise 5: Show why this follows.

It follows from what we have just said that if there is a "true" probability distribution for English, it will assign a *lower* probability to any given corpus than the empirical distribution based on that corpus, and that the empirical distribution based on one corpus  $C_1$  will assign a lower probability to a different corpus  $C_2$  than  $C_2$ 's own empirical distribution. Putting that in terms of entropy (that is, taking the positive log of the probabilities that we have just mentioned, and dividing by  $N$ , the number of words in the corpus), we may say that the "true" probability distribution for English assigns a *larger* entropy to a corpus  $C$  than  $C$ 's own empirical distribution, and that  $C_1$ 's empirical distribution assigns a higher entropy to a different corpus  $C_2$  than  $C_2$ 's own empirical distribution does.

These notions are so important that some names have been applied to these concepts. When we calculate this formula, weighting *one* distribution (like an observed frequency distribution) by the log probabilities of some other distribution  $D_2$ , we call that the *cross-entropy*; and if we calculate the difference between the cross-entropy and the usual (self) entropy, we also say that we are calculating the Kullback-Leibler (or "KL") divergence between the two distributions. Mathematically, if the probability assigned to word<sub>*i*</sub> by  $D_1$  is expressed as  $D_1(\text{word}_i)$  (and likewise for  $D_2$ ), then the KL divergence is

$$\sum_{j=1}^V D_1(\text{word}_j) \log D_1(\text{word}_j) - D_1 \log D_2(\text{word}_j)$$

The tricky part is being clear on why  $D_1$  appears before the log in both terms in this equation. It is because there, the  $D_1$  is being used to indicate how many times (or what proportion of the time) this particular word occurs in the corpus we are looking at, which is entirely separate from the role played by the distribution inside the log function -- *that* distribution tells us what probability to assign to the given word. (Solomon Kullback and Richard Leibler were among the original mathematicians at the National Security Agency.)

The KL divergence just above can be written equivalently as

$$\sum_{j=1}^V D_1(\text{word}_j) \log \frac{D_1(\text{word}_j)}{D_2(\text{word}_j)}$$

A common notation for this is:  $KL(D_1||D_2)$ . Note that this relationship is not symmetric:  $KL(D_1 || D_2)$  is not equal to  $KL(D_2 || D_1)$ .

Here's one direct application of these notions to language. Suppose we have a set of letter frequencies (forming *distributions*, of course) from various languages using the Roman alphabet. For purposes of this illustration, we'll assume that whatever accents the letters may

have had in the original, all letters have been ruthlessly reduced to the 26 letters of English. Still, each language has a different set of frequencies for the various letters of the alphabet, and these various distributions are called  $D_i$ . If we have a sample from one of these languages with empirical distribution  $S$  (that is, we *count* the frequencies of the letters in the sample), we can algorithmically determine which language it is taken from by computing the KL divergence  $KL(S||D_i)$ . The distribution which produces the lowest KL divergence is the winner -- it is the correct language, for its distribution best matches that of the sample.

### **Conditional probabilities, this time with logs**

---

We have talked about the *conditional probability* of (for example) a word  $w$ , given its left-hand neighbor  $v$ , and we said that we can come up with an empirical measure of it as the total number of  $v+w$  biwords, divided by the total number of  $v$ 's in the corpus:

$$p(S[i] = w | S[i-1] = v) = \frac{p(vw)}{p(v)} . \text{ Look at the log-based version of this:}$$

$$\log p(S[i] = w | S[i-1] = v) = \log p(vw) - \log p(v) .$$

### **Essential Information Theory**

---

Suppose we have given a large set of data from a previously unanalyzed language, and four different analyses of the verbal system are being offered by four different linguists. Each has an account of the verbal morphology using rules that are (individually) of equal complexity. There are 100 verb stems. Verbs in each group use the same rules; verbs in different groups use entirely different rules.

Linguist 1 found that he had to divide the verbs into 10 groups with 10 verbs in each group.

Linguist 2 found that she had to divide the verbs into 10 groups, with 50 in the first group, 30 in the second group, 6 in the third group, and 2 in each of 7 small groups.

Linguist 3 found that he had just one group of verbs, with a set of rules that worked for all of them.

Linguist 4 found that she had to divide the verbs into 50 groups, each with 2 stems in it.

Rank these four analyses according how good you think they are -- sight unseen.

Hopefully you ranked them this way:

Best: Linguist 3

Linguist 2

Linguist 1

Worst: Linguist 4

And why? Because the entropy of the sets that they created goes in that order. That's not a coincidence -- entropy measures our intuition of the degree of organization of information.

The entropy of a set is  $-\sum p(a_i) \log p(a_i)$ , where we sum over the probability of each subset making up the whole -- and where the *log* is the base-2 log.

- The entropy of Linguist 1's set of verbs is  $-1 * 10 * 1/10 * \log (1/10) = \log (10) = 3.32$ .
- The entropy of Linguist 2's set of verbs is  $-1 * (1/2 \log (1/2) + 0.3 * \log (0.3) + 0.06 * \log (0.06) + 0.14 * \log (0.02)) = 0.346 + 0.361 + 0.169 + 0.548 = 1.42$ .
- The entropy of Linguist 3's set of verbs is  $-1 * 1 * \log (1) = 0$ .
- The entropy of Linguist 4's set of verbs is  $-1 * 50 * 1/50 * \log (0.02) = 3.91$ .

Thus, in some cases – very interesting ones, in my opinion – the concept of entropy can be used to quantify the notion of elegance of analysis.

### **Another approach to entropy**

---

The traditional approach to explaining information and entropy is the following. A language can be thought of as an organized way of sending symbols, one at a time, from a sender to a receiver. Both have agreed ahead of time on what the symbols are that are included. How much *information* is embodied in the sending of any particular symbol?

Suppose there are 8 symbols that comprise the language, and that there is no bias in favor of any of them – hence, that each of the symbols is equally likely at any given moment. Then sending a symbol can be thought of as being equivalent to being willing to play a yes/no game – essentially like a child's Twenty Questions game. Instead of receiving a symbol passively, the receiver asks the sender a series of yes/no questions until he is certain what the symbol is. The number of questions that is required to do this – on average – is the average information that this symbol-passing system embodies.

The best strategy for guessing one of the 8 symbols is to ask a question along the lines of "Is it one of symbols 1, 2, 3, or 4?" If the answer is Yes, then ask "Is it among the set: symbols 1 and 2?" Clearly only one more question is needed at that point, while if the answer to the first question is No, the next question is, "Is it among the set: symbols 5 and 6?" And clearly only one more question is needed at that point.

If a set of symbols has  $N$  members in it, then the best strategy is to use each question to break the set into two sets of size  $N/2$ , and find out which set has the answer in it. If  $N = 2^k$ , then it will take  $k$  questions; if  $N = 2^k + 1$ , it will take  $k+1$  questions.

Note that if we did all our arithmetic in base 2, then the number of questions it would take to choose from  $N$  symbols would be no more than the number of digits in  $N$  (and occasionally it takes 1 fewer).  $8 = 1000_2$ , and it takes 3 questions to select from 8 symbols;  $9 = 1001_2$ , and it takes 4 questions to select from 9 symbols;  $15 = 1111_2$ , and it takes 4 questions to select from 15 symbols.

Summarizing: the amount of information in a choice from among N possibilities (possible symbols, in this case) is  $\log N$  bits of information, rounding up if necessary. Putting it another way -- if there are N possibilities, and they each have the same probability, then each has probability  $1/N$ , and the number of bits of information per symbol is the positive log probability (which is the same thing as the log of the reciprocal of the probability).

Exercise 6: why is the positive log probability the same thing as the log of the reciprocal of the probability?

But rarely is it the case that all of the symbols in our language have the same probability, and if the symbols have different probabilities, then the average number of yes/no questions it takes to identify a symbol will be less than  $\log N$ . Suppose we have 8 symbols, and the probability of symbol 1 is 0.5, the probability of symbol 2 is 0.25, and the probability of the other 6 is one sixth of the remaining 0.25, i.e.,  $1/24$  each.

In this case, it makes sense to make the first question be simply, "Is it Symbol #1?" And half the time the answer will be "yes". If the answer is "No," then the question could be, "Is it Symbol #2?" and again, half the time the answer will be "Yes." Therefore, in three-fourths of the cases, the average number of questions needed will be no greater than 2. For the remaining six, let's say that we'll take 3 more questions to identify the symbol.

So the average number of questions altogether is  $(0.5 * 1) + (0.25 * 2) + (0.25 * 5) = 0.5 + 0.5 + 1.25 = 2.25$ . (Make sure you see what we just did.) When the probabilities are not uniformly distributed, then we can find a better way to ask questions, and the better way will lower the average number of questions needed.

All of this is a long, round-about way of saying that the average information per symbol decreases when the probabilities of the symbols is not uniform. This quantity is the *entropy* of the message system, and is the weighted average of the number of bits of information in *each symbol*, which obeys the generalization mentioned just above: the information is  $-1$  times the log of the probability of the symbol, i.e., the positive log probability.

The entropy is, then:

$$\sum \text{prob}(x_i) \log\left(\frac{1}{\text{prob}(x_i)}\right) = - \sum \text{prob}(x_i) \log(\text{prob}(x_i))$$

## Mutual information

---

Mutual information is an important concept that arises in the case of a sample space consisting of joint events: each event can be thought of as a pair of more basic events. One possible example would be the input and the output of some device (like a communication channel); another, very different example could be successive letters, or successive words, in a corpus. Let's consider the case of successive words.

The joint event, in this case, is the occurrence of a *biword* (or bigram, if you prefer). "of the" is such an event; so is "the book", and so on. We can compute the entropy of the set of all the bigrams in a corpus. We can also consider the separate events

that constitute the joint event: e.g., the event of "the" occurring as a left-hand member of a biword. That, too, has an observed frequency, and so we can compute its entropy -- and of course, we can do that for the right-hand words of the set of bigrams. We want to know what the relationship is between the entropy of the joint events and the entropy of the individual events.

If the two words comprising a biword are statistically unrelated -- independent -- then the entropy of the joint event is the sum of the entropies of the individual events. We'll work through that, below. But linguistically, we know that this won't in fact be the case. If you know the left-hand word of a bigram, then you know a lot about what is likely to be the right-hand word: that is to say, the entropy of the possible right-hand words will be significantly *lower* when you know the left-hand word. If you know that the left-hand word is *the*, then there is an excellent chance that the right-hand word is *first*, *best*, *only* (just look at Table 2 above!). The entropy of the words in Table 2 is much lower than the entropy of the whole language. This is known as the *conditional entropy*: it's the entropy of the joint event, *given* the left-hand word constant. If we compute this conditional entropy (i.e., right-hand word entropy based on *knowing* the left-hand word) for all of the left-hand words of the biword, and take the weighted mean of these entropies, what you have computed is called the *mutual information*: it is an excellent measure of how much knowledge of the first word tells you about the second word (and this is true for any joint events).

Mutual information between two random variables  $i, j$ :

$$(MM-1) \quad \sum_i p(x_i) \sum_j -p(y_j | x_i) \log p(y_j | x_i)$$

$$(While\ we're\ at\ it,\ don't\ forget\ that\ p(y_j|x_i) = \frac{p(x_i y_j)}{p(x_i)} = \frac{p(bigram)}{p(word)} )$$

It is certainly not obvious, but the following is true: if you compute the conditional entropy of the *left*-hand word, given the right-hand word, and compute the weighted average over all possible right-hand words, you get the same quantity, the mutual information. Mutual information is symmetric, in that sense.

There is a third way of thinking about mutual information which derives from the following, equivalent formula for mutual information:

$$(MM-3) \quad \sum p(x_i y_j) \log \frac{p(x_i y_j)}{p(x_i) p(y_j)},$$

where  $p(x_i)$  is the probability of  $x_i$ , which is to say,  $\sum_j p(x_i, y_j)$ . This last expression, (MM-3), can be paraphrased as: the weighted difference between the information of the joint events (on the one hand) and the information of the separate events (on the other). That is, if the two events were independent, then  $\frac{p(x_i y_j)}{p(x_i) p(y_j)}$  would be 1.0, and its log of that would be zero.

So far, all of our uses of mutual information have been weighted averages (we often find it handy to refer to this kind of average as an ensemble average, which borrows a metaphor from statistical physics). However, in computational linguistics applications,

it is often very useful to compute  $\log \frac{p(x_i y_j)}{p(x_i) p(y_j)}$  for *individual* bigrams. The most straightforward way to use it is to compare the log probability assigned to a string of words under two models: (1) a unigram model, in which each word is assigned a (positive) log probability (remember this formula from above?  $\sum_{i=1}^N \{ \text{prob}(S[i]) \}$

and (2) a bigram model, in which each word is assigned a positive log probability, conditioned by its left-hand neighbor.  $\sum_{i=1}^N \log p(S[i] = w_i \mid S[i-1] = w_{i-1})$ . (You have to do something special to deal with the probability of the first word here.) It turns out, as we shall see, that the *difference* between the sum of the logs of the unigram model and the sum of the logs on the bigram model is just the sum of the mutual information between the successive pairs of words. That is, the difference in the goodness of a bigram and a unigram model is the (sum of the) mutual information between successive words.

STOP. Think about that.

This is because the bigram model gives us:

$$\sum_{i=1}^N \log p(S[i] = w_i \mid S[i-1] = w_{i-1}) = \sum_{i=1}^N \log p(w_{i-1} w_i) - \log p(w_{i-1})$$

$$\text{while the MI is } \sum_{i=1}^N \log p(w_{i-1} w_i) - \log p(w_{i-1}) - \log p(w_i)$$

Again: bigram model = unigram model + mutual information.