Proceedings of the 10th Seminar of the International Association for Tibetan Studies,
Oxford, United Kingdom, Sept. 6-12, 2003 — Information Technology Panel

An Entropy-based Assessment of the Unicode Encoding for Tibetan

Paul G. Hackett
Columbia University
New York, New York, U.S.A.
ph2046@columbia.edu

Abstract

This paper presents an analysis of the Unicode encoding scheme for Tibetan from the standpoint of morpheme entropy. We can speak of two levels of entropy in Tibetan: syllable entropy (a measure of the probability of the sequential occurrence of syllables), and morpheme entropy (a measure of the probability of the sequential occurrence of characters or morphemes), the latter being a measure of the redundancy of the language. Syllable entropy is a purely statistical calculation that is a function of the domain of the literature sampled, while morpheme entropy, we show, is relatively domain independent given a statistically significant sample. Morpheme entropy can be calculated statistically, though a theoretical upper bound can also be postulated based on language dependent morphology rules. This paper presents both theoretical and statistical estimates of the morpheme entropy for Tibetan, and explores the Tibetan Unicode encoding scheme in relation to data compression, and other issues analyzed in light of entropy-based language modeling.

Introduction

Although derived from the context of statistical mechanics and thermodynamics, the concept of entropy was introduced to the realm of information theory by Claude Shannon who defined it simply as a measure of the uncertainty or amount of disorder in a system. Shannon postulated the Fundamental Coding Theorem that stated that the lower bound to the average number of bits per symbol needed to encode a message was given by its entropy. Since the entropy of a system relates to the unpredictability of a data point in a sequence given a previous data point, if the entropy of a system can be reduced, then the predictability of the next data point increases. The key to decreasing entropy and increasing the predictability of elements in a sequence is contingent on the representation (i.e., language model) of the system.

In 1948, Claude Shannon published "A Mathematical Theory of Communication"[1] in which he discussed the uncertainty or amount of disorder of a communications system. From the set of axioms that he proposed to model this behavior, he identified a quantity $\mathcal{H}$, which he called entropy. Analogous to its role in a thermodynamics context, Entropy can also be considered as a measure of randomness in a system, and can be utilized both to verify the accurate arrival of messages and as a mechanism for reducing the physical size of messages while retaining their meaning.

In constructing various language models, there are a number of perceived advantages to the construction of a minimal entropy model:

- the language model that represents a data sequence in a manner that lowers the overall entropy without compromising its representational power is a more efficient medium;[2]
- the entropy of a language is a lower bound for the compression ratio of any compression algorithm for linguistic data, and thus a language model so constructed approaches this limit;
- the theoretical entropy value represents a measure of the complexity of the language script.

In this paper we explore the calculation and implications of both the theoretical and statistical measures of the entropy of the Tibetan language.

The Information Measure of Symbol Sequences

In many information systems, not every symbol is equally likely to be used in a given communication. Looking at the English language for example, the letter "e" is 12 times more likely to occur than other letters in a statistically representative text sample.[3] This uneven distribution of symbols in a language is also a characteristic of distinctive groups of letters or words (*n*-grams).[4]

When extended to the ASCII symbol set, it is immediately obvious that each of the 256 symbols in ASCII is not likely to occur an equal number of times in any meaningful communication (with the obvious, trivial exception of a communication consisting of the ASCII set). Since digital communications occur in the medium of binary values (1 and 0), if a message is encoded in binary in the most efficient manner possible, then the average number of binary digits ("bits") required per symbol of the source language is given by the entropy. Hence, the entropy of a set of equally likely symbols (such as the digits 0–9 in a table of random numbers) is simply the logarithm (base 2) of the number of symbols in the set, or in this case, $\log_2(10) = 2.30$ bits per symbol. The entropy of the English alphabet — which contains 27 case-insensitive characters (26 letters and a space) — is similarly: $\log_2(27)$ or a theoretical upper limit of 4.76 bits per symbol. Since meaningful communications in the English language tend not to consist of random sequences of characters, but rather contain an uneven distribution of them, the actual entropy of the English language is considerably lower, and is calculated as follows:

$$\mathcal{H} = -(P_1\log_2 P_1 + P_2\log_2 P_2 + \dots P_i\log_2 P_i + \dots + P_{27}\log_2 P_{27})$$

Where, $\mathcal{P}_i$ is the probability of occurrence of the i^{th} character in the English alphabet, and the symbol $\mathcal{H}$ is the entropy. Simplified, the algorithm is:

$$\mathcal{H} = -\sum_{i=1}^{n} \mathcal{P}_i \log_2 \mathcal{P}_i$$

Where, n represents the n possible distinct characters or symbols in a language or code. Since the probability, $\mathcal{P}_i$ of any given symbol (i.e., letter) x_i is contingent upon the previous letter(s), the contingent (bi-gram) probability function takes the form of:

$$\mathcal{P}_i = -\sum_{j=1}^{n} \mathcal{P}(x_i | x_j)$$

with the obvious extrapolation to an n-gram model. Performing statistical calculations for a $n = 8$ model, Shannon calculated the entropy of printed English to be roughly 2.3 bits per symbol; carrying the calculation up to $n = 100$, he estimated the entropy to be 1.3 bits per symbol. The English language, he concluded was roughly 75 % redundant in its formal representation.[5]

The Entropy Calculation for Literary Tibetan

Applying these calculations to the Tibetan language, we can compute upper-bound estimates for the entropy of printed Tibetan. In the traditional presentation of the Tibetan language — omitting ornamental flourishes and foreign words — there are forty-four discrete morphemes: thirty letters, four explicit vowels, three super-scripts, four sub-scripts, a syllable delimiter (*tsheg*), a phrase delimiter (*shad*), and whitespace. Assuming a simple even distribution ($\mathcal{P}_i = 1/44$) yields an upper-bound for the morpheme entropy of Tibetan of $\mathcal{H} = 5.46$ bits per symbol.

Unlike English, however, which has a small number of generic letter combination rules for normative text ("q" followed by "u" etc.), Tibetan possesses a finite set of morphological constraint rules for syllable formation. These rules are summarized and illustrated in Table 1 and Figure 1.

Prefix

ག ད བ མ འ

Superscript

ར ལ ས
⊗ ⊗ ⊗

Root Letters

ཀ ཁ ག ང ཅ ཆ ཇ ཉ ཏ ཐ ད ན པ ཕ བ མ ཙ ཚ ཛ ཝ ཞ ཟ འ ཡ ར ལ ཤ ས ཧ ཨ

Subscripts

⊗ ⊗ ⊗ ⊗
ྭ ྱ ྲ ླ

Vowels

⊗ི ⊗ུ ⊗ེ ⊗ོ

Suffix

ག ང ད ན བ མ འ ར ལ ས

Secondary Suffix

ས

Table 1. Morpheme Topology Restrictions

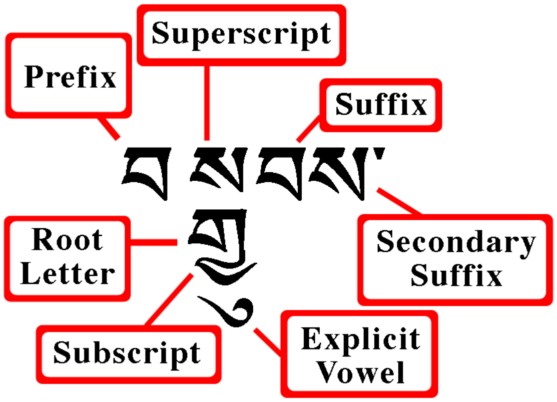

Fig. 1. Morpheme Classes in an Example Tibetan Syllable

Ignoring non-Tibetan constructions (Sanskrit, etc.), it is possible to calculate the morpheme entropy, $\mathcal{H}$, of Tibetan. Assuming an even distribution over topologically-constrained morphemes as described above (Table 1) in combination with traditional morpheme restriction rules, it is possible to recalculate probability values according to an order-1 finite context language model (i.e., conditional bi-grams) where the probability $\mathcal{P}_i$ is given by:

$$\mathcal{P}_i = -\sum_{j=1}^{44} \mathcal{P}(x_i \mid x_j)$$

This yields an entropy of $\mathcal{H}$ = 4.82 bits per symbol (Figure 2).

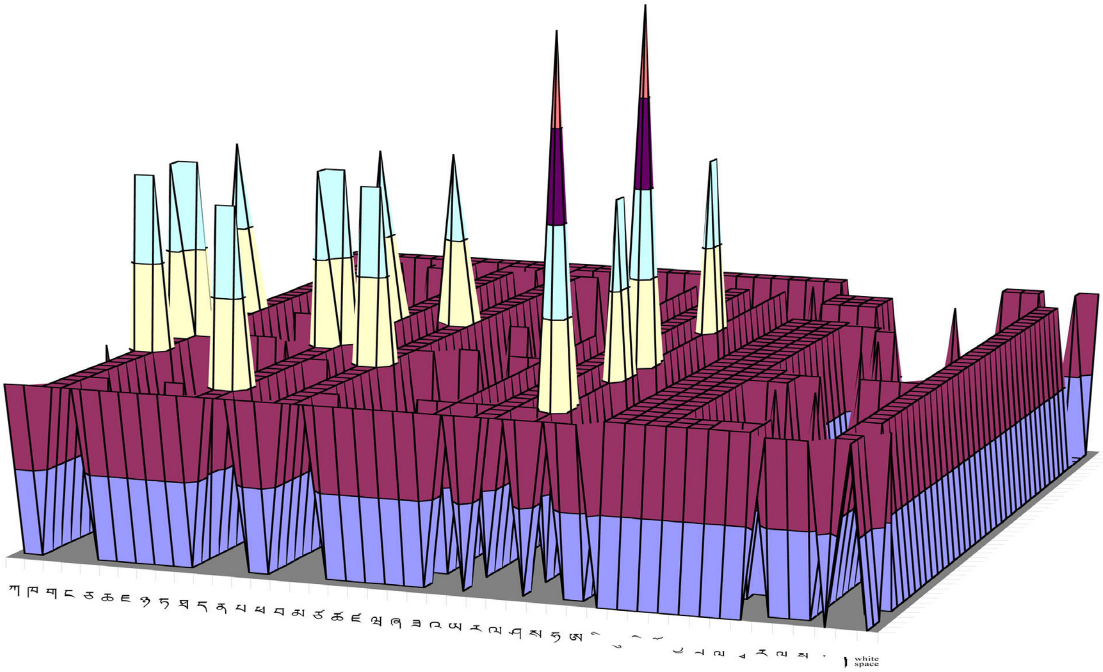

Fig. 2. Bi-gram Probabilities — 44 Morphemes (Theoretical)

Unicode

The Tibetan encoding within Unicode was first proposed in December 1990[6] and based on a Sanskritic model relying on a virāma. This version, although structurally similar to character encoding model used for many of the Indic scripts, was subsequently deprecated and removed. The foundation for the current Unicode Tibetan scheme reappeared in altered form in version 2.0.0 of the standard (July 1996) and encoded all morphemes given in Table 1 (above) except the three superscripts. In this encoding, the three superscripts were conflated with their full height equivalent characters, and a second set of zero-width, subjoined Tibetan characters was added. The question of whether or not the three superscripts are the same morphemes[7] or characters[8] as their full height equivalent characters — since different positions within an individual syllable convey different phonemic and collational information — was debated at length. For the sake of systematic processing and encoding, however, the decision was made to define these superscripts as the same characters as their full height equivalent characters (although they do appear to be different morphemes) and to map them to the same code points. One of the issues addressed in this paper is whether or not this distinction is statistically significant in terms of the entropy of the language model.

In its currently established form, the Unicode encoding for Tibetan does not employ the traditional morpheme distinctions. Rather, this standard gives preference to a "stack-friendly" encoding (suitable for keyboard interface design), which necessitates back-tracking to properly disambiguate the three superscript morphemes from traditional "root" letters. This has the unfortunate side-effect of complicating the implementation of traditional sort rules for Tibetan,[9] but the advantage of simplifying "stack" construction for a rendering engine.

In the Unicode scheme, fifty-six characters are required to completely specify the same set of data as the traditional forty-four given above. These fifty-six characters consist of the thirty letters (filling the prefix, root, superscript, and two suffix positions), nineteen subjoined consonants (representing root letters which take a superscript and the four traditional subscripts), four explicit vowels, two logical delimiters for syllable and phrase boundaries, and whitespace (Table 2).

Root Letters and "Superscripts"

ཀ ཁ ག ང ཅ ཆ ཇ ཉ ཏ ཐ ད ན པ ཕ བ མ ཙ ཚ ཛ ཝ ཞ ཟ འ ཡ ར ལ ཤ ས ཧ ཨ

Subscripts and Subscribed Root Letters

⊗ྐ ⊗ྒ ⊗ྔ ⊗ྕ ⊗ྗ ⊗ྙ ⊗ྟ ⊗ྡ ⊗ྣ ⊗ྤ ⊗ྦ ⊗ྨ ⊗ྩ ⊗ྫ ⊗ྭ ⊗ྱ ⊗ྲ ⊗ླ ⊗ྷ

Vowels Delimiters

⊗ི ⊗ུ ⊗ེ ⊗ོ “ ” ་ །

Table 2. Unicode characters needed to represent normative Tibetan

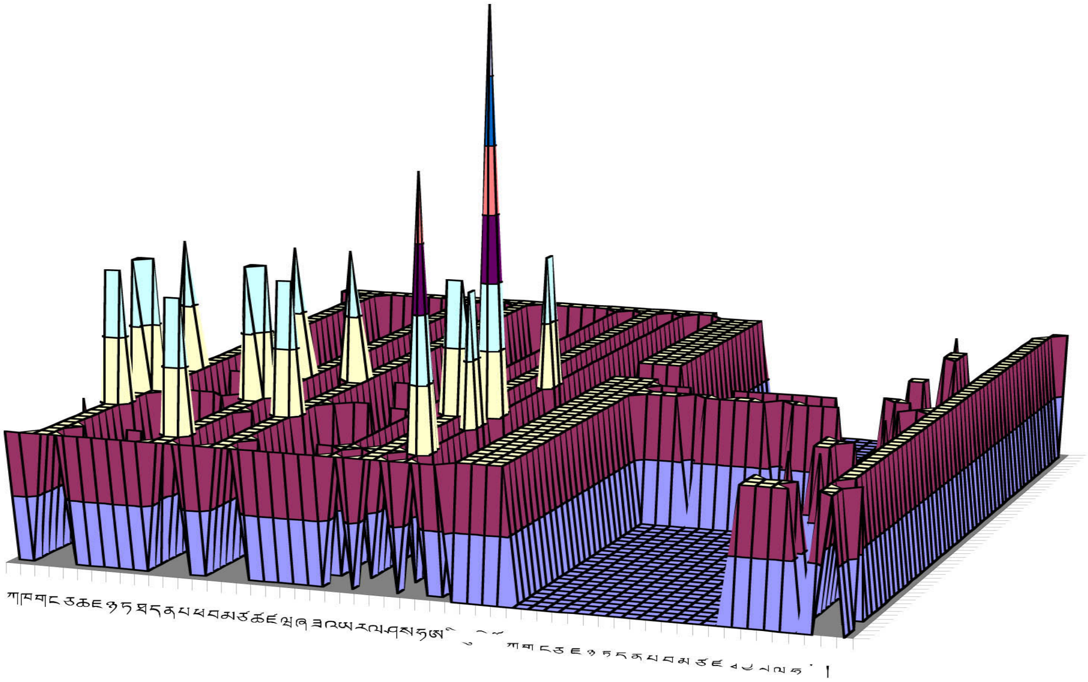

Fig. 3. Bi-gram Probabilities — 56 Morphemes (Theoretical)

A simple even distribution ($\mathcal{P}_i$ = 1/56) yields an upper-bound for the morpheme entropy of Tibetan of $\mathcal{H}$ = 5.81 bits per symbol, but assuming an even distribution over topologically-constrained characters based on traditional character restriction rules with probabilities derived from an order-1 finite context language model yields a morpheme entropy of $\mathcal{H}$ = 4.79 bits per symbol (Figure 3).

It is possible that this measure could vary when all Sanskritic letter combinations are accounted for. For example, in his design of the Tibetan encoding for the "Comparative Kangyur and Tengyur" (*bka' bstan dpe bsdur ma*) project, Tashi Tsering identified over 7,000 Sanskritic character stacks.[10] Although many are rare, some occurring only once in the canon, their existence must be allowed for in any encoding and compression scheme. Nonetheless, there is no statistically significant difference between the two language models.

Statistical Calculations of Morpheme Entropy

As can be seen, both the traditional and Unicode encodings yield similar estimates of the morphological entropy of Tibetan, roughly 4.8 bits per symbol. A statistical estimate was calculated using the Unicode encoding with an order-1 finite context language model over the ACIP data set. The bi-gram probabilities (Figure 4) yield an estimate of the entropy at $\mathcal{H}$ = 4.35 bits per symbol.

The first observation, immediately apparent in Figure 4, is that the bi-gram probabilities are heavily dominated by syllable boundary effects, that is, the characters that appear adjacent to syllable delimiters (*tsheg*) and phrase delimiters (*shad*). This can be seen when the contingent (bi-gram) probabilities are collapsed into simple probabilities. Figure 5 shows a comparison between the theoretical and statistically derived probabilities for the individual morphemes.

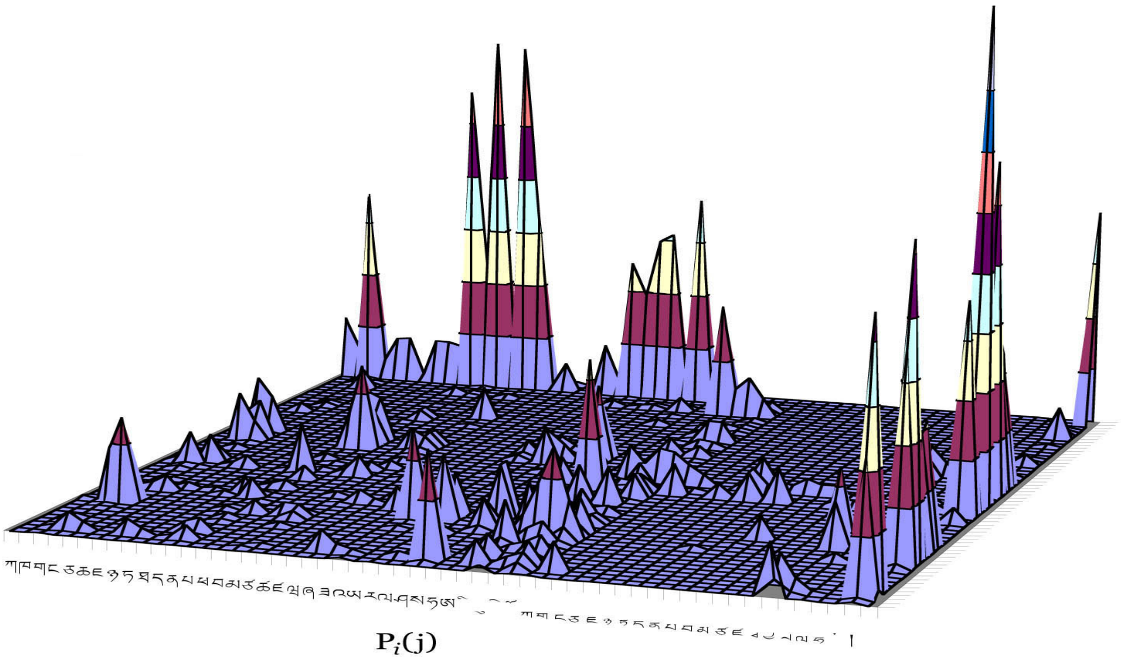

Fig. 4. Bi-gram Probabilities — 56 Morphemes (Statistical)

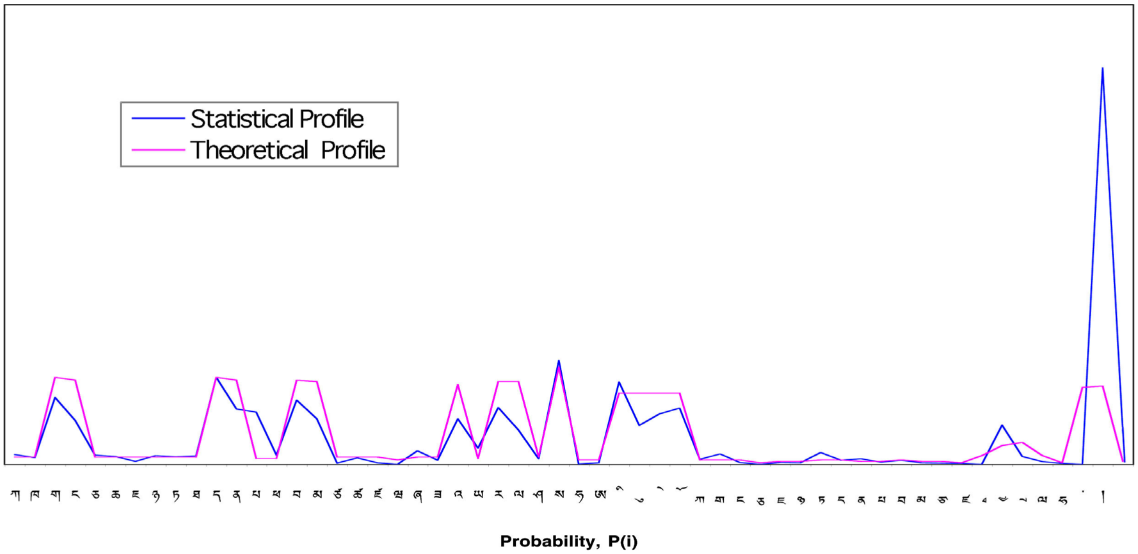

Fig. 5. Theoretical vs. Statistical Probabilities

A second observation — also apparent from Figure 5 — is that excluding syllable boundary effects, the statistical probabilities for letter bi-grams follow their theoretical values closely. Taking into account language features that are outside the scope of a bi-gram language model — the fact that the average length of a Tibetan syllable is five characters, and the average length of a *shad*-delimited phrase is sixty characters (i.e., twelve syllables) — the theoretical probabilities can be appropriately re-weighted and the resultant entropy calculated. The result is a theoretical morpheme entropy value of $\mathcal{H} = 4.37$ bps, which agrees closely with the statistical calculation, and hence appears to be a valid approximation of the morpheme entropy of literary Tibetan.

Variable Compression Experiments

While the above experiments were conducted on a statistically large sample, it is possible to compute the morpheme entropy for smaller, more homogeneous texts and text collections. For example, the statistical entropy values for a subset of texts and a few individual texts are given in Table 3.

Sample	Entropy, H (bps)
twenty Bka'-'gyur texts	4.29
Diamond Sutra	4.17
Candrakīrti's Prasannapadā	3.78

Table 3. Entropy Values for Limited samples of literary Tibetan texts

Hence, in practical applications, a variable compression rate can be achieved for some individual texts that is considerably lower than both the theoretical and statistical averages, though quickly approaches the statistical limit with as few as twenty texts.

An Entropy-based Data Compression Algorithm

Most text compression algorithms are tailored for Euro-American languages. In examining the performance of these for non-Euro-American languages, Vines and Zobel found that no currently publicly available text compression techniques performed very well for Chinese, and their application to that language required either excessive memory or yielded only moderate compression.[11] Tibetan, being an alphabetic language however, does not suffer from the same problems as Chinese does. Hence, any generic entropy compression routine would yield a usable degree of compression for either transmission or storage.

Furthermore, it has been shown that although Tibetan is an unsegmented language possessing syllable- and phrase-delimiters with no explicit word boundaries, by invoking a small number of grammar rules a shallow parser can yield 95% of all word boundaries. The result is that in retrieval experiments comparing words against *n*-grams, the average length of a Tibetan word was determined to be two syllables. Consequently, a bi-gram probability function would be sufficient to approximate syllable-level entropy for Tibetan.

Although other constraint rules could be brought to bear in $n > 2$ morpheme entropy calculations, given the observed dominance of syllable boundary probabilities, it is more likely that any gains in compression would be a reflection of syllable-level effects rather than a refined morpheme-level language model.

Final Remarks

This paper represents an initial foray into Tibetan language entropy research as a foundation for additional inquiry. While a morphologically-constrained bi-gram language model offers a working upper limit for the letter entropy of literary Tibetan ($\mathcal{H} = 4.8$ bps), it is insufficient to truly approximate a statistical estimate. When the complexity of the language model is slightly increased by the incorporation of syllable and phrase boundary probabilities however, the theoretical estimate and statistical estimate converge. As a result, we estimate morpheme entropy for literary Tibetan at $\mathcal{H} = 4.4$ bps.

Although the significance of compression routines has been lessened of late in light of substantial progress in storage media and communications bandwidth, language entropy research remains valuable in terms of other secondary applications beyond text compression such as spell-checking, data corruption recovery, and OCR error detection. In addition, extrapolations of this research into larger n-gram language models including syllable n-grams offers potential as a parser evaluation metric and as a tool for neologism detection.

Notes

1. Claude Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, 27 [1948]: 379-423, 623-656.
2. Caroline Lyon, and Bob Dickerson, *Exploiting statistical characteristics of word sequences for the efficient coding of speech*. Technical report. Computer Science Department, University of Hertfordshire (1999).
3. Gilbert Held, *Understanding Data Communications*. NY: Addison Wesley (2000), p.93.
4. A two symbol (or word) pair can be referred to as a bi-gram, similarly tri-grams, etc.; an indexable sequence of n symbols is generically referred to as a set of n-grams. Common high frequency letter bi-grams in the English language, for example, are "th" and "er."
5. The immediate implication of Shannon's calculation being that it would be theoretically possible to reconstruct a message even if every other letter were lost or corrupted. Claude Shannon, "Prediction and entropy of printed English," *Bell System Technical Journal*, 30 [1951]: 50-64.
6. The Unicode Consortium, *Unicode 1.0*. Draft Standard, n.p., December 1990.
7. That is, "a minimally distinctive unit of writing in the context of a particular writing system." *The Unicode Standard, Version 3.0*.
8. That is, "the abstract meaning and/or shape, rather than a specific shape." *The Unicode Standard, Version 3.0*.
9. For example, a problem with conflation of ར་ and ར་མགོ་, etc. is that only (a minimum) $n = 3$ (tri-gram) probability calculation would sufficiently disambiguate the occurrence of one from the other. Similarly, the character bi-grams involving n are insufficiently distinguished in the instances: བས་, བསམ་, བསྣ, and between ་ས, ་ས, where expectation values would shift radically between root, superscript, and suffix usages. Root and suffix uses are sufficiently determined by a bi-gram probability, though root vs. superscript usage is not, and the conflation results in either a greater entropy for a low n-value Unicode language model, or necessitating a higher

n-value language model. For a discussion of Tibetan sort rules under Unicode, refer to: Robert Chilton, "Sorting Unicode Tibetan using a Multi-Weight Collation Algorithm," Paper presented at the Tenth International Association for Tibetan Studies (IATS-X) Conference, Oxford, United Kingdom, September 6-12, 2003.

10. private communication.

11. Phil Vines and Justin Zobel, "Compression Techniques for Chinese Text," p.1300.

References

Chilton, Robert, "Sorting Unicode Tibetan using a Multi-Weight Collation Algorithm," Paper presented at the Tenth International Association for Tibetan Studies (IATS-X) Conference, Oxford, United Kingdom, September 6-12, 2003.

Hackett, Paul G. *Information Retrieval for Tibetan: Segmentation vs. n-grams*. Masters Thesis. University of Maryland – College Park (2000).

Hansel, Georges, Dominique Perrin, and Imre Simon. "Compression and Entropy" in (Alain Finkel, Matthias Jantzen, eds.) *STACS 92, 9th Annual Symposium on Theoretical Aspects of Computer Science*. (1992), pp.515-528.

Held, Gilbert. *Understanding Data Communications*. NY: Addison Wesley (2000).

Lyon, Caroline. "Evaluating Parsing Schemes with Entropy Indicators," paper presented at the Fifth Meeting on Mathematics of Language - MOL5 (1997). [http://www.dfki.de/events/mol/]

Lyon, Caroline, and Bob Dickerson. *Exploiting statistical characteristics of word sequences for the efficient coding of speech*. Technical report. Computer Science Department, University of Hertfordshire (1999). [http://citeseer.ist.psu.edu/article/lyon99exploiting.html]

Shannon, Claude. "A Mathematical Theory of Communication," *Bell System Technical Journal*, 27 [1948]: 379-423, 623-656.

Shannon, Claude. "Prediction and entropy of printed English," *Bell System Technical Journal*, 30 [1951]: 50-64.

Unicode Consortium. *Unicode 1.0*. Draft Standard, n.p., December 1990.

Unicode Consortium. *Unicode Standard, Version 3.0*. Reading, MA: Addison-Wesley (2000).

Vines, Phil, and Justin Zobel. "Compression Techniques for Chinese Text," *Software: Practice and Experience* 28(12) [1998]: 1299-1314.