Estimating and Comparing Entropy across Written Natural Languages Using PPM Compression

Frederic H. Behr, Jr.* Victoria Fossum † Michael Mitzenmacher‡
David Xiao§

Abstract

Previous work on estimating the entropy of written natural language has focused primarily on English. We expand this work by considering other natural languages, including Arabic, Chinese, French, Greek, Japanese, Korean, Russian, and Spanish. We present the results of PPM compression on machine-generated and human-generated translations of texts into various languages.

Under the assumption that languages are equally expressive, and that PPM compression does well across languages, one would expect that translated documents would compress to approximately the same size. We verify this empirically on a novel corpus of translated documents. We suggest as an application of this finding using the size of compressed natural language texts as a mean of automatically testing translation quality.

1 Introduction

Accurately estimating the entropy of written natural language has great practical importance for a variety of applications in information theory and language modeling. Most directly, entropy provides a theoretical lower bound and a target for compression. Similarly, entropy provides a guide for language modeling; a language model should accurately reflect the entropy of the underlying language. Accurate language models are important in a variety of areas, including speech recognition, handwriting recognition, spell-checking, etc.

Estimating the entropy of the English language therefore unsurprisingly has a long history in the information theory literature. Since the true probability distribution of symbols in English is unknown, the entropy of English cannot be computed directly. Instead, one can approximate the probability distribution of symbols by some probability model and compute the cross-entropy, which provides an upper bound on the true entropy of the source. Early experiments in estimating the entropy of written English by Shannon, and later by Cover and King, estimated the probability distribution of English

*Harvard College. behr@fas.harvard.edu

†University of Michigan. vfossum@eecs.umich.edu

‡Contact author. Harvard University, Div. of Engineering and Applied Sciences. 33 Oxford Street, Cambridge, MA 02138. 617-496-7172. michaelm@eecs.harvard.edu

§Harvard College. dxiao@fas.harvard.edu

by measuring the ability of human subjects to predict the next character in a body of text. Shannon estimated an entropy between 0.6 and 1.3 bits per character (bpc)[10]; Cover and King estimated 1.25 bpc [4]. Later experiments in estimating entropy have used machines to measure the performance of compression algorithms on English text, based on the concept that an efficient compression algorithm can closely approximate the true entropy of a source. In particular, the PPM compression algorithm appears to compress English text quite effectively. The lowest estimates of entropy produced via compression have been achieved using a variation of PPM by Teahan and Cleary, who report estimates of 1.46 bpc on Dumas Malone's *Jefferson the Virginian*, the same text used by Shannon in his human experiments [13].

We extend prior work on estimating the entropy of English text by comparing the entropy of each of the following written languages: Arabic, Chinese, English, French, Greek, Japanese, Korean, Russian, and Spanish. Using PPM as an approximation to an ideal compression algorithm, we compress both human- and machine-generated translations of a variety of written texts in each of these languages. We have several motivations for this exercise. One is purely technical; it is interesting to have these numbers for various languages, especially languages with alphabets and character sets very different from English, such as the Asian languages.

But perhaps our primary motivation is to answer the following natural thought experiment. Under the assumption that languages are equally expressive, and that PPM compression does similarly well across languages, one would expect that translated documents would compress to approximately the same size. If this were not the case, it would suggest that either languages are fairly disparate in expressiveness, violating conventional wisdom in linguistics [5], or the success of PPM compression is language-specific, either of which would be interesting results.[1] Previous studies of natural languages indicate that they share many statistical similarities, perhaps the most famous of which is Zipf's law, which says that word frequency follows a power-law distribution [14, 6]. The structure of most languages follow other well-formed similar patterns [5]. One such pattern bearing particular relevance to the comparative performance of a Markov-based compression algorithm such as PPM across languages is the universal tendency for semantically related words to appear next to each other, sequentially, in a sentence [1]. Prior work motivates our suggestion that the information content of a text should be similar no matter what language it is written in, and hence that translated texts should compress to approximately the same size.

We verify that translated documents do appear to have the same information content empirically on a novel corpus of translated documents. We believe that this cross-language corpus is itself one of the contributions of this work; the corpus will be made publicly available.

We suggest as an application of this finding using the size of compressed natural language texts as a means of automatically testing translation quality. Machine-generated translations have been growing substantially with the advent of business on the World Wide Web. Anecdotal experience relating poor translation quality abounds; automatic tests that can help spot poor translations before they are used could help in determining

[1] A third alternative would be that the translations are extremely poor; more on this below.

which texts require more human input. While humans can spot good and poor translations fairly easily, automating the task appears to be challenging. There are examples of work in this area [7, 9], but the problem is not yet adequately solved. Our work follows the spirit of program checking [3], which involves trying to find methods to check a program that are computationally less expensive than the original program. Our suggestion is that compressing the original translated texts and performing a size comparison may provide a tool for catching poor translations. Before presenting our methodology and experimental results, we outline the thought experiment that motivates this application below.

2 Compressing Translations: A Thought Experiment

We suggest that the compressed size of texts with the same information content should remain close to constant across languages, even when the uncompressed texts vary in size. That is, the number of bits required to encode a particular text should be independent of the encoding and language used.

Our hypothesis is based on the following intuition. As stated above, the estimates of the entropy of English are based on a finite stochastic model of the language. The relevant attributes of these models can be applied to all natural languages. The first is the set of *statements* that can be expressed in this language. Technically, statements are simply characters strings; however, the loose concept of meaning is meant to be embodied by this informal term.

$$S^L = \{S_i^L : S_i^L \text{ is an statement that can be expressed in the language}\}$$

Our conclusions rely on the assumption that S^L is the same for all natural languages. In other words, all natural languages are equally expressive. This assumption appears backed by a commonly held belief in linguistics theory, which states that all natural languages possess grammars and lexicons that are rich enough to express and communicate any conceivable thought [8, 5]. Over this set, we have a probability distribution describing the likelihood that a statement is expressed, or output by the source.

$$p^L = \{(S_i^L, p_i^L) : p_i^L \text{ is the probability that } S_i^L \text{ is expressed in this language } L\}$$

We believe that this distribution will be extremely similar across languages. This is not to say that every statement has the same probability in each language; for example, "I am speaking English" appears more likely to be spoken in English. Furthermore, we do not presume that all languages are equally capable of encoding a particular statement using the same optimal number of bits. Statements containing highly specialized lexical subsets of a language (such as those written in jargon) could result in considerable variation in the length of the encoding of the statement when expressed in different languages. For example, a user's manual for an automobile might suffer considerable bloating in the number of bits required to transmit the information contained when the manual undergoes translation from its original language to a language spoken by an agricultural people without automobiles. We believe that when a sufficiently broad

collection of statements is considered, discrepancies in the expressive power of different languages with regard to their ability to encode a particular statement efficiently will tend to even out.

As a whole, therefore, for large samples of statements, the probability distributions for different languages are likely to be quite similar. Given the set S^L and the probability distribution p^L an optimal encoding of ideas can be determined; standard information theory says that the approximate length ℓ_i^L for a given statement S_i^L should be:

$$\ell_i^L = \lceil \log_2 \frac{1}{p_i^L} \rceil$$

This length, ℓ_i^L, is what we are approximating empirically using compression; with a good compression algorithm we expect to come close to this size. If our assumptions that p^L is roughly the same across all languages is true, we would expect compressed translations to have approximately the same size. Again, if this is not true, this says something interesting about linguistic variation across languages.

3 Methodology

While our results yield potentially interesting bits-per-symbol statistics for various languages, for comparison purposes we compare the total length of compressed texts.

We present results for both human- and machine-generated translations of texts. As human-generated translations are time-consuming and expensive to produce, we limited our experiments to texts with existing translations available electronically in multiple languages. In our experiments we used the Bible, which we obtained in Arabic, Chinese, English, French, Spanish, Korean, Japanese, and Russian.

The other large human-translated corpus that we used is a set of treaties from the United Nations Treaty Collection with translations available in English, Spanish, French, Chinese, Arabic, and Russian. The collection contains nineteen documents in all. All of them are present in English, Spanish, French, and Chinese. Eighteen are present in Russian, and eleven are present in Arabic. These texts are suitable for our experiments for two reasons. First, as these documents were prepared by the UN, the translations are presumably extremely accurate. Second, a treaty should, in theory, have a very exact and literal meaning, and should therefore be the same in each language. On the negative side, the legal style of the English texts seems more disjointed than standard prose, especially after preprocessing.

We generated machine translations of the Bible using the commercially available Systran PRO Premium 3.0 translation software. Systran translation software is used for example in the translation engine Altavista Babelfish.

We performed some preprocessing of the texts in each language to standardize their format before analysis. It has been customary in previous work in estimating the entropy of written English to convert all letters to uppercase and delete any characters other than letters and spaces, leaving a twenty-seven character alphabet [10][13]. We performed this conversion for English and similarly filtered the other languages. Furthermore, while English text can be represented using the standard ASCII encoding table, other

Language	Encoding	Language	Encoding
English	ASCII	French	ISO-8859-1
Spanish	ISO-8859-1	Chinese	EUC-CN (GB2312)
Korean	EUC-KR	Japanese	EUC-JP
Arabic	Windows CP-1256	Russian	ISO-8859-5

Table 1: Encodings used for various languages.

languages require additional characters. We used a common encoding for each language. For example, French and Spanish texts were first converted to the ISO-Latin-1 8-bit character set, then filtered. The resulting alphabet of the French texts included 15 accented characters beyond the twenty-seven character alphabet for English, while that of the Spanish texts included 6 accented characters beyond the English alphabet. For Chinese, we used texts in the GB character set with EUC-CN encoding, in which each Chinese ideograph is represented by a two-byte sequence. Our Chinese text alphabet consists of all Chinese characters, without any spaces or other non-character data. See Table 3 for a complete listing of the encodings we used for each language.

Our baseline was the PPMD+ compression algorithm as implemented by Teahan [12]. Our experiments also used a Linux port of Charles Bloom's variation of PPM called PPMZ [2, 11]. PPMZ is essentially an improved version of PPMD+ using more efficient escaping mechanisms and local order estimation, and has empirically outperformed PPMD+.

For both algorithms, it has been shown that using similar texts to train the PPM model results in significantly better results. We indicate below the training that was done for each different set of texts.

4 Experimental Results

Our results are described below. For each set of texts, we report the following measurements: the text's original size, its compressed size, the ratio of the text's original size to the original size of its English translation, and the ratio of the text's compressed size to the compressed size of its English translation. Since we claim that the size of a given compressed text should be similar across languages, we expect the ratio between compressed sizes to be close to 1.

4.1 Results for the Bible

We split the Bible into training and testing sets as follows: the first 20 books, from Genesis to Proverbs, were used to train the PPM model, then the remaining books were compressed using the trained model to obtain the above results. This setup produced reasonable results for the English text, the compressed text contained 1.62 bits per character in the original.

As Table 2 shows, all ratios between compressed text sizes in different languages become substantially closer to 1 after compression; the compressed text sizes are within

Language	Original Size (bytes)	Ratio (Original)	Compressed Size (bytes)	Ratio (Compressed)
English	1936473	1	390846	1
Spanish	1804756	0.932	384681	0.962
French	1896459	0.979	393805	1.01
Chinese	884860	0.457	337505	0.864
Korean	1259920	0.651	346478	0.886
Arabic	1875204	0.968	418443	1.071
Japanese	1519224	0.785	452337	1.157
Russian	1506920	0.778	376162	0.962

Table 2: Results for the Bible using the PPMD compression algorithm. The ratio is the ratio of the size in the language divided by the size in English. We expect the ratio of the compressed sizes to be close to 1.

roughly 15% of English. Note that the languages deviating the most from English in original size, namely Arabic, Chinese, and Korean, are between 3-10 times closer to English in size after compression. The other languages besides Japanese are close to English in size both before and after compression. We remark further on the Japanese translation below.

As we suggested previously, there are three possible reasons we might expect deviation from the ideal ratio of 1: poor translations, insufficiently powerful compression algorithms, or differences in the expressiveness of languages. Some of this is likely due to the fact that the books are not strictly translations, in the sense that the Spanish version was derived directly from this specific English text; they are simply both representations of the Bible. We initially conjectured that there is still some variation due to the difference in compression performance across languages.

To support this conjecture, we further experimented with PPMZ, which should yield better compression. The results of PPMZ on the same Bible corpus appear in Table 3. As can be seen, PPMZ indeed yields better compression for most languages, the apparent exceptions being Chinese and Korean. Overall, the ratios are significantly closer to 1 under these experiments.

The only consistently surprising outcome is with the Japanese translation, which performs poorly under both compression algorithms. We speculate on a possible cause. The Japanese Bible source we used initially had editorial comments; we tried to remove as many of these comments as possible. However, because none of the authors is fluent in Japanese, there may be additional extraneous content remaining in the text we compressed. Hence, we believe this is an example of a poor translation, in that extraneous text appears. This is a demonstration that comparing the compressed size can potentially be a useful tool for finding poor translations, a point we elaborate on below.

Note that the Bible comprises sixty-six different books, each with its own unique style and subject matter. For the most part, the books were written independently by different authors. This diversity of writing style within the text can cause problems for adaptive compression algorithms such as PPM, which struggle with changing contexts

Language	Original Size (bytes)	Ratio (Original)	Compressed Size (bytes)	Ratio (Compressed)
English	1936473	1	363288	1
Spanish	1804756	0.932	360535	0.992
French	1896459	0.979	359903	0.991
Chinese	884860	0.457	341850	0.941
Korean	1259920	0.651	352440	0.970
Arabic	1875204	0.968	395242	1.09
Japanese	1519224	0.785	438135	1.206
Russian	1506920	0.778	362207	0.997

Table 3: Results for the Bible using the PPMZ compression algorithm

Language	Original Size (bytes)	Ratio (Original)	Compressed Size (bytes)	Ratio (Compressed)
English	1936473	1	411732	1
Spanish	1804756	0.932	414206	1.006
French	1896459	0.979	422532	1.026
Chinese	884860	0.457	370168	0.899
Korean	1259920	0.651	387532	0.941
Arabic	1875204	0.968	448962	1.090
Japanese	1519224	0.785	481649	1.170
Russian	1506920	0.778	412354	1.002

Table 4: Results for the Bible using the BZIP2 compression algorithm

and styles because the probability models they build may not reflect the underlying probability distribution of the text immediately after a shift in context.

Finally, we note that as a check of these results, we have also compressed these texts using gzip and bzip2. Neither compression scheme achieves the compression results of the PPM algorithms, and hence the entropy estimates for the languages are necessarily less accurate using these compression schemes. It is worthwhile noting, however, that in comparing the ratios, the same basic trends are apparent, including the behavior of the Japanese text.

4.2 Results for the United Nations Treaties

Results for the UN treaties vary somewhat. We see similar results for Arabic and Chinese; the compressed sizes are much closer than before compression, and very close to the ideal ratio of 1. The other languages, however, do not seem to do as well. They all compress to around 10-15% larger than English. We cannot determine a clear explanation for this; however, we suspect that the legal nature and jargon included in the text may not be amenable to translation, potentially causing the deviations. While the ideas contained in the Bible are likely to be universal in that they can be readily

Language	Original Size (bytes)	Ratio (Original)	Compressed Size (bytes)	Ratio (Compressed)
English	1936473	1	562720	1.000
Spanish	1804756	0.932	559261	0.994
French	1896459	0.979	572904	1.018
Chinese	884860	0.457	438738	0.780
Korean	1259920	0.651	510311	0.907
Arabic	1875204	0.968	627727	1.116
Japanese	1519224	0.785	654144	1.162
Russian	1506920	0.778	532343	0.946

Table 5: Results for the Bible using the GZIP compression algorithm

Language	Original Size (bytes)	Ratio (Original)	Compressed Size (bytes)	Ratio (Compressed)
English	977885	1	110848	1
Spanish	1064225	1.088	123695	1.116
French	1050979	1.075	125214	1.130
Chinese	420178	0.430	109279	0.986
English	944778	1	106501	1
Russian	1009347	1.068	127750	1.200
English	478026	1	65310	1
Arabic	328606	0.687	69350	0.942

Table 6: Results for the UN Treaties using the PPMZ compression algorithm

lexicalized in a variety of languages, the ideas expressed in the UN treaties belong to a specialized subset of language which can result in the generation of long explanations of word meanings in the target language after translation.

Hence, this may again be an example of poor translations, or it may be an instance where certain statements (say legal statements) are indeed more probable in certain languages than in others. This remains an interesting point of study for future work.

4.3 Machine Translation

We performed machine translation of the KJV Bible into a variety of European languages using Systran software. We note that the results here also do not show the ideal ratio of 1; instead, the ratios of the compressed sizes are closer to the original sizes.

We notice from Table 7 that all of the machine-translated texts are in general larger than we would expect, which is not the case with human-translated versions of the Bible. Again, we have found one possible reason for this that is a flaw in the translation process. When the translation software does not recognize an English word, it simply outputs the English word directly into the translated text. Unfortunately, this occurs rather often because of the KJV's use of archaic English vocabulary and conjugation (e.g. 'gaveth'

Language	Original Size (bytes)	Ratio (Original)	Compressed Size (bytes)	Ratio (Compressed)
English	1936473	1	390846	1
French	2150962	1.111	432672	1.107
Spanish	2048227	1.058	406895	1.041
German	2113502	1.091	422693	1.081
Italian	2097506	1.083	439109	1.123

Table 7: Results for the Machine Translated Bible using the PPMZ compression algorithm

instead of 'gave'). This causes a possible complication in building the probability model for PPM because the algorithm must take into account vocabulary from both languages.

Again, this test highlights a possible use of compression to detect poor translations: a ratio between the compressed text size before and after translation that is substantially far from 1 may be a means of detecting poor translations. This experiment provides an example, in that the ratios larger than 1 appear to be due to a flaw in the translation process.

We emphasize that a ratio close to 1 does not necessarily imply a faithful translation. (Indeed, one could achieve a ratio of 1 with our test by doing no translation at all!) The compression-based test we propose is therefore one-sided, in that it can detect poor translations, but it can only be considered as auxiliary evidence that a translation is good. Designing additional automatic tests for determining fidelity in translation remains a topic for further work.

5 Conclusion

We extend prior work on estimating the entropy of English text by using PPM compression results to compare the entropy of each of the following written languages: Arabic, Chinese, English, French, Greek, Japanese, Korean, Russian, and Spanish. We suggest a direct relationship between the size of compressed natural language texts and the information content of those texts that is independent of the text's language and encoding. Our initial tests, although preliminary, provide some support to our conjecture that translation preserves information content. Based on this conjecture, we suggest compression as a possible means for detecting poor translations.

We believe that our work opens the way for future work involving compression and translation across languages. An important question is whether current compression techniques are biased toward languages with a Roman alphabet. Although most compression schemes are designed to work for general sources, in practice specifics of the language may affect performance. Our initial work suggests that PPM techniques perform well across languages, but of course PPM is rather expensive computationally in practice. Another question is whether there is a more suitable corpus available. We suspect, for example, that there may be a more representative corpus than the UN documents we have used here; extending our work to include a wider sample of genres

would be a worthwhile exercise. More generally, we have asked how one might check machine translations automatically, using fewer resources than the translation itself requires. Compression offers a possible means of finding poor translations; we believe this idea can be developed further, to find for example smaller sections of the text that may be translated poorly.

References

[1] Mark C. Baker. *The Atoms of Language*. Basic Books, United States of America, first edition, 2001.

[2] C. Bloom. Solving the problems of context modeling, 1998. www.cbloom.com/papers/ppmz.zip.

[3] M. Blum and S. Kannan. Designing programs that check their work. In *Proc. 21st ACM Symposium on the Theory of Computing*, pages 86–97, 1989.

[4] T. Cover and R. King. A convergent gambling estimate of the entropy of English. *IEEE Transactions on Information Theory*, 24(4):413–421, 1978.

[5] V. Fromkin and R. Rodman. *An Introduction to Language*. Harcourt Brace College Publishers, Fort Worth, sixth edition, 1998.

[6] B. Mandelbrot. An informational theory of the statistical structure of languages. In Betterworth W. Jackson, editor, *Communication Theory*, pages 486–502, 1953.

[7] S. Niessen, F. Och, G. Leusch, and N. Hermann. An evaluation tool for machine translation: Fast evaluation for mt research. In *Proc. 2nd International Conference on Language Resources and Evaluation*, pages 39–45, Athens, Greece, May 2000.

[8] William O'Grady, Michael Dobrovolsky, and Mark Aronoff. *Contemporary Linguistics: An Introduction*. New York: St. Martin's Press, 1989.

[9] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation, 2001. Published as IBM Report RC22176.

[10] C. Shannon. Prediction and entropy of printed English. *Bell Systems Technical Journal*, 30:50–64, 1951.

[11] J. Tarhio and H. Peltola. PPMZ for Linux. www.cs.hut.fi/u/tarhio/ppmz/ppmz.tar.gz.

[12] W. Teahan. PPMD. ftp.cs.waikato.ac.nz/pub/compression/ppm/ppm.tar.gz.

[13] W. J. Teahan and John G. Cleary. The entropy of English using PPM-based models. In *Data Compression Conference*, pages 53–62, 1996.

[14] G. Zipf. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, Cambridge, MA, 1932.