A Computational Approach to Deciphering Unknown Scripts

Kevin Knight
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
knight@isi.edu

Kenji Yamada
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
kyamada@isi.edu

Abstract

We propose and evaluate computational techniques for deciphering unknown scripts. We focus on the case in which an unfamiliar script encodes a known language. The decipherment of a brief document or inscription is driven by data about the spoken language. We consider which scripts are easy or hard to decipher, how much data is required, and whether the techniques are robust against language change over time.

1 Introduction

With surprising frequency, archaeologists dig up documents that no modern person can read. Sometimes the written characters are familiar (say, the Phoenician alphabet), but the language is unknown. Other times, it is the reverse: the written script is unfamiliar but the language is known. Or, both script and language may be unknown.

Cryptanalysts also encounter unreadable documents, but they try to read them anyway. With patience, insight, and computer power, they often succeed. Archaeologists and linguists known as *epigraphers* apply analogous techniques to ancient documents. Their decipherment work can have many resources as input, not all of which will be present in a given case: (1) monolingual inscriptions, (2) accompanying pictures or diagrams, (3) bilingual inscriptions, (4) the historical record, (5) physical artifacts, (6) bilingual dictionaries, (7) informal grammars, etc.

In this paper, we investigate computational approaches to deciphering unknown scripts, and report experimental results. We concentrate on the following case:

- unfamiliar script
- known language
- minimal input (monolingual inscriptions only)

This situation has arisen in many famous cases of decipherment—for example, in the Linear B documents from Crete (which turned out to be a "non-Greek" script for writing ancient Greek) and in the Mayan documents from Mesoamerica. Both of these cases lay unsolved until the latter half of the 20th century (Chadwick, 1958; Coe, 1993).

In computational linguistic terms, this decipherment task is not really translation, but rather text-to-speech conversion. The goal of the decipherment is to "make the text speak," after which it can be interpreted, translated, etc. Of course, even after an ancient document is phonetically rendered, it will still contain many unknown words and strange constructions. Making the text speak is therefore only the beginning of the story, but it is a crucial step.

Unfortunately, current text-to-speech systems cannot be applied directly, because they require up front a clearly specified sound/writing connection. For example, a system designer may create a large pronunciation dictionary (for English or Chinese) or a set of manually constructed character-based pronunciation rules (for Spanish or Italian). But in decipherment, this connection is unknown! It is exactly what we must discover through analysis. There are no rule books, and literate informants are long-since dead.

2 Writing Systems

To decipher unknown scripts, is useful to understand the nature of known scripts, both ancient and modern. Scholars often classify scripts into three categories: (1) alphabetic, (2) syllabic,

Figure 1: The Phaistos Disk (c. 1700BC). The disk is six inches wide, double-sided, and is the earliest known document printed with a form of movable type.

and (3) logographic (Sampson, 1985).

- Alphabetic systems attempt to represent single sounds with single characters, though no system is "perfect." For example, Semitic alphabets have no characters for vowel sounds. And even highly regular writing systems like Spanish have plenty of spelling variation, as we shall see later.
- Syllabic systems have characters for entire syllables, such as "ba" and "shu." Both Linear B and Mayan are primarily syllabic, as is Japanese kana. The Phaistos Disk from Crete (see Figure 1) is thought to be syllabic, because of the number of distinct characters present.
- Finally, logographic systems have characters for entire words. Chinese is often cited as a typical example.

Unfortunately, actual scripts do not fall neatly into one category or another (DeFrancis, 1989; Sproat, forthcoming). Written Japanese will contain syllabic kana, alphabetic roomaji, and logographic kanji characters all in the same document. Chinese characters actually have a phonetic component, and words are often composed of more than one character. Irregular English writing is neither purely alphabetic nor purely logographic; it is sometimes called morphophonemic. Ancient writing is also mixed, and archaeologists frequently observe radical writing changes in a single language over time.

3 Experimental Framework

In this paper, we do not decipher any ancient scripts. Rather, we develop algorithms and apply them to the "decipherment" of known, modern scripts. We pretend to be ignorant of the connection between sound and writing. Once our algorithms have come up with a proposed phonetic decipherment of a given document, we route the sound sequence to a speech synthesizer. If a native speaker can understand the speech and make sense of it, then we consider the decipherment a success. (Note that the native speaker need not even be literate, theoretically). We experiment with modern writing systems that span the categories described above. We are interested in the following questions:

- Can automatic techniques decipher an unknown script? If so, how accurately?
- What quantity of written text is needed for successful decipherment? (this may be quite limited by circumstances)
- What knowledge of the spoken language is needed? Can it to be extracted automatically from available resources? What quantity of resources?
- Are some writing systems easier to decipher than others? Are there systematic differences among alphabetic, syllabic, and logographic systems?
- Are word separators necessary or helpful?
- Can automatic techniques be robust against language evolution (e.g., modern versus ancient forms of a language)?
- Can automatic techniques identify the language behind a script as a precursor to deciphering it?

4 Alphabetic Writing (Spanish)

Five hundred years ago, Spaniards invaded Mayan lands, burning documents and effectively eliminating everyone who could read and write. (Modern Spaniards will be quick to point out that most of the work along those lines

Sounds:

B, D, G, J (ny as in canyon), L (y as in yarn), T (th as in thin), a, b, d, e, f, g, i, k, l, m, n, o, p, r, rr (trilled), s, t, tS (ch as in chin), u, x (h as in hat)

Characters:

ñ, á, é, í, ó, ú, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

Figure 2: Inventories of Spanish sounds (with rough English equivalents in parentheses) and characters.

had already been carried out by the Aztecs). Mayan hieroglyphs remained uninterpreted for many centuries. We imagine that if the Mayans have invaded Spain, then 20th century Mayan scholars might be deciphering ancient Spanish documents instead.

We begin with an analysis of Spanish writing. The task of decipherment will be to re-invent these rules and apply them to written documents in reverse. First, is necessary to settle on the basic inventory of sounds and characters. Characters are easy; we simply tabulate the distinct ones observed in text. For sounds, we need something that will serve as reasonable input to a speech synthesizer. We use a Spanish-relevant subset of the International Phonetic Alphabet (IPA), which seeks to capture all sounds in all languages. Actually, we use an ASCII version of the IPA called SAMPA (Speech Assessment Methods Phonetic Alphabet), originally developed under ESPRIT project 1541. There is also a public-domain Castillian speech synthesizer (called Mbrola) for the Spanish SAMPA sound set. Figure 2 shows the sound and character inventories.

Now to spelling rules. Spanish writing is clearly not a one-for-one proposition:

- a single sound can produce a single character (a → a)
- a sound can produce two characters (tS → c h)
- two sounds can produce a single character (k s → x)

B → b or v
D → d
G → g
J → ñ
L → l l or y
a → a or á
b → b or v
d → d
e → e or é
f → f
g → g
i → i or í
l → l
m → m
n → n
o → o or ó
p → p
r → r
t → t
tS → c h
u → u or ú
x → j
nothing → h
T (followed by a, o, or u) → z
T (followed by e or i) → c or z
T (otherwise) → c
k (followed by e or i) → q u
k (followed by s) → x
k (otherwise) → c
rr (at beginning of word) → r
rr (otherwise) → rr
s (preceded by k) → nothing
s (otherwise) → s

Figure 3: Spanish sound-to-character spelling rules. The left-hand side of each rule contains a Spanish sound (and possible conditions), while the right-hand side contains zero or more Spanish characters.

- silence can produce a character (h)

Moreover, there are ambiguities. The sound L (English y-sound) may be written as either ll or y. The sound i may also produce the character y, so the pronunciation of y varies according to context. The sound rr (trilled r) is written rr in the middle of a word and r at the beginning of a word.

Figure 3 shows a sample set of Spanish

spelling rules. We formalized these rules computationally in a finite-state transducer (Pereira and Riley, 1997). The transducer is bidirectional. Given a specific sound sequence, we can extract all possible character sequences, and vice versa. It turns out that while there are many ways to write a given Spanish sound sequence with these rules, it is fairly clear how to pronounce a written sequence.

In our decipherment experiment, we blithely ignore many of the complications just described, and pretend that Spanish writing is, in fact, a one-for-one proposition. That is, to write down a sound sequence, one replaces each sound with a single character. We do allow ambiguity, however. A given sound may produce one character sometimes, and another character other times.

Decipherment is driven by knowledge about the spoken language. In the case of archeological decipherment, this knowledge may include vocabulary, grammar, and meaning. We use simpler data. We collect frequencies of sound-triples in spoken Spanish. If we know that triple "t l k" is less frequent than "a s t," then we should ultimately prefer a decipherment that contains the latter instead of the former, all other things being equal.

This leads naturally into a statistical approach to decipherment. Our goal is to settle on a sound-to-character scheme that somehow maximizes the probability of the observed written document. Like many linguistic problems, this one can be formalized in the noisy-channel framework. Our sound-triple frequencies can be turned into conditional probabilities such as P(t | a s). We can estimate the probability of a sound sequence as the product of such local probabilities.

$$P(s_1 \ldots s_n) \sim P(s_3 \mid s_1\ s_2) \cdot P(s_4 \mid s_2\ s_3) \cdot P(s_5 \mid s_3\ s_4) \cdot \ldots$$

A specific sound-to-character scheme can be represented as a set of conditional probabilities such as P(v | B). Read this as "the probability that Spanish sound B is written with character v." We can estimate the conditional probability of entire character sequence given an entire sound sequence as a product of such probabilities.

$$P(c_1 \ldots c_n \mid s_1 \ldots s_n) \sim P(c_1 \mid s_1) \cdot P(c_2 \mid s_2) \cdot P(c_3 \mid s_3) \cdot \ldots$$

Armed with these basic probabilities, we can compute two things. First, the total probability of observing a particular written sequence of characters $c_1 \ldots c_n$:

$$P(c_1 \ldots c_n) = \sum_{s_1 \ldots s_n} P(s_1 \ldots s_n) \cdot P(c_1 \ldots c_n \mid s_1 \ldots s_n)$$

And second, we can compute the most probable phonetic decipherment $s_1 \ldots s_n$ of a particular written sequence of characters $c_1 \ldots c_n$. This will be the one that maximizes $P(s_1 \ldots s_n \mid c_1 \ldots c_n)$, or equivalently, maximizes $P(s_1 \ldots s_n) \cdot P(c_1 \ldots c_n \mid s_1 \ldots s_n)$. The trick is that the P(character | sound) probabilities are unknown to us. We want to assign values that maximize $P(c_1 \ldots c_n)$. These same values can then be used to decipher.

We adapt the EM algorithm (Dempster et al., 1977), for decipherment, starting with a uniform probability over P(character | sound). That is, any sound will produce any character with probability 0.0333. The algorithm successively refines these probabilities, guaranteeing to increase $P(c_1 \ldots c_n)$ at each iteration. EM requires us to consider an exponential number of decipherments at each iteration, but this can be done efficiently with a dynamic-programming implementation (Baum, 1972). The training scheme is illustrated in Figure 4.

In our experiment, we use the first page of the novel Don Quixote as our "ancient" Spanish document $c_1 \ldots c_n$. To get phonetic data, we might tape-record modern Spanish speakers and transcribe the recorded speech into the IPA alphabet. Or we might use documents written in an alternate, known script, if any existed. In this work, we take a short cut by reverse-engineering a set of medical Spanish documents, using the finite-state transducer described above, to obtain a long phonetic training sequence $s_1 \ldots s_n$.

At each EM iteration, we extract the most probable decipherment and synthesize it into audible form. At iteration 0, with uniform probabilities, the result is pure babble. At iteration 1, Spanish speakers report that "it sounds like someone speaking Spanish, but using words I don't know." At iteration 15, the decipherment can be readily understood. (Recordings can be accessed on the World Wide Web at `http://www.isi.edu/natural-`

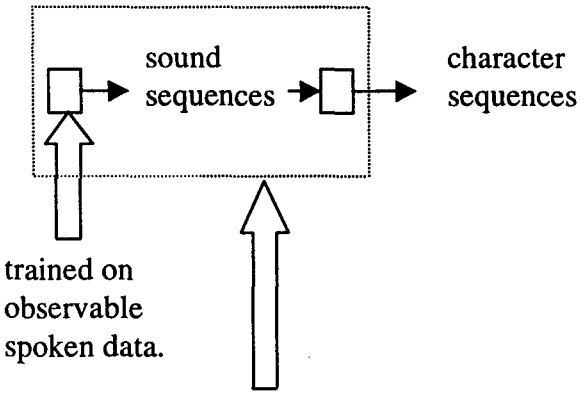

Figure 4: Training scheme for decipherment. We first train a phonetic model on phonetic data. We then combine the phonetic model with a generic (uniform) spelling model to create a probabilistic generator of character sequences. Given a particular character sequence ("ancient document"), the EM algorithm searches for adjustments to the spelling model that will increase the probability of that character sequence.

`language/mt/decipher.html`).

If we reverse-engineer Don Quixote, we can obtain a gold standard phonetic decipherment. Our automatic decipherment correctly identifies 96% of the sounds. Incorrect or dropped sounds are due to our naive one-for-one model, and not to weak algorithms or small corpora. For example, "de la Mancha" is deciphered as "d e l a m a n T i a" even though the characters ch really represent the single sound tS rather than the two sounds T i.

Figure 5 shows how performance changes at each EM iteration. It shows three curves. The worst-performing curve reflects the accuracy of the most-probable decipherment using the formula above, i.e., the one that maximizes $P(s_1 \ldots s_n) \cdot P(c_1 \ldots c_n \mid s_1 \ldots s_n)$. We find that it is better to ignore the $P(s_1 \ldots s_n)$ factor altogether, because while the learned sound-to-

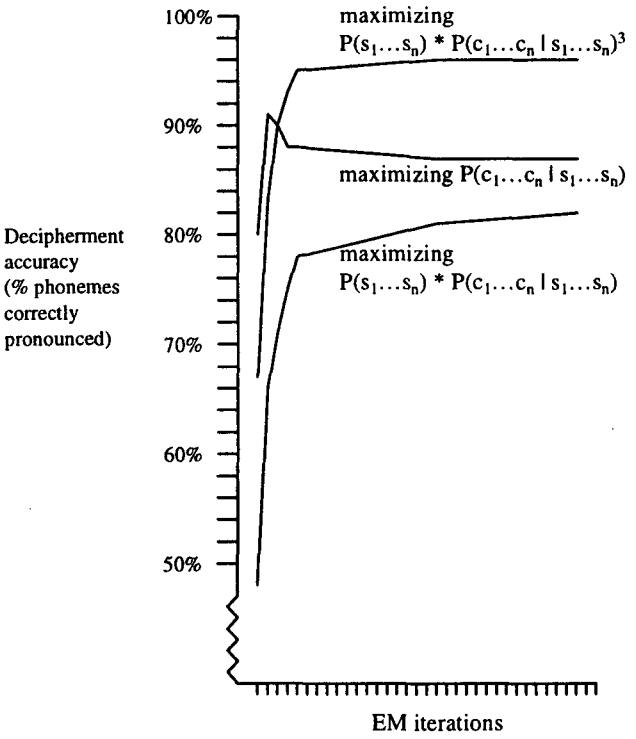

Figure 5: Performance of Spanish decipherment. As we increase the number of EM iterations, we see an improvement in decipherment performance (measured in terms of correctly generated phonemes). The best result is obtained by weighting the learned spelling model more highly than the sound model, i.e., by choosing a phonetic decoding $s_1 \ldots s_n$ for character sequence $c_1 \ldots c_n$ that maximizes $P(s_1 \ldots s_n) \cdot P(c_1 \ldots c_n \mid s_1 \ldots s_n)^3$.

character probabilities are fairly good, they are still somewhat unsure, and this leaves room for the phonetic model to overrule them incorrectly. However, the $P(s_1 \ldots s_n)$ model does have useful things to contribute. Our best performance, shown in the highest curve, is obtained by weighing the learned sound-to-character probabilities more highly, i.e., by maximizing $P(s_1 \ldots s_n) \cdot P(c_1 \ldots c_n \mid s_1 \ldots s_n)^3$.

We performed some alternate experiments. Using phoneme pairs instead of triples is workable—it results in a drop from 96% accuracy to 92%. Our main experiment uses

word separators; removing these degrades performance. For example, it becomes more difficult to tell whether the r character should be trilled or not. In our experiments with Japanese and Chinese, described next, we did not use word separators, as these languages are usually written without them.

5 Syllabic writing (Japanese Kana)

The phonology of Japanese adheres strongly to a consonant-vowel-consonant-vowel structure, which makes it quite amenable to syllabic writing. Indeed, the Japanese have devised a *kana* syllabary consisting of about 80 symbols. There is a symbol for ka, another for ko, etc. Thus, the writing of a sound like K depends on its phonetic context. Modern Japanese is not written in pure kana; it employs a mix of alphabetic, syllabic, and logographic writing devices. However, we will use pure kana as a stand-in for wide range of syllabic writing systems, as Japanese data is readily available. We obtain kana text sequences from the Kyoto Treebank, and we obtain sound sequences by using the finite-state transducer described in (Knight and Graehl, 1998).

As with Spanish, we build a spoken language model based on sound-triple frequencies. The sound-to-kana model is more complex. We assume that each kana token is produced by a sequence of one, two, or three sounds. Using knowledge of syllabic writing in general—plus an analysis of Japanese sound patterns—we restrict those sequences to be (1) consonant-vowel, (2) vowel-only, (3) consonant-no-vowel, and (4) consonant-semivowel-vowel. For initial experiments, we mapped "small kana" onto their large versions, even though this leads to some incorrect learning targets, such as KIYO instead of KYO. We implement the sound- and sound-to-kana models in a large finite-state machine and use EM to learn individual weights such as P(ka-kana | SH Y U). Unlike the Spanish case, we entertain phonetic hypotheses of various lengths for any given character sequence.

Deciphering 200 sentences of kana text yields 99% phoneme accuracy. We render the sounds imperfectly (yet inexpensively) through our public-domain Spanish synthesizer. The result is comprehensible to a Japanese speaker.

We also experimented with deciphering smaller documents. 100 sentences yields 97.5% accuracy; 50 sentences yields 96.2% accuracy; 20 sentences yields 82.2% accuracy; five sentences yields 48.5% accuracy. If we were to give the sound sequence model some knowledge about words or grammar, the accuracy would likely not fall off as quickly.

6 "Logographic" writing (Chinese)

As we mentioned in Section 2, Chinese characters have internal phonetic components, and written Chinese does not really have a different character for every word: so, it is not really logographic. However, it is representative of writing systems whose distinct characters are measured in the thousands, as opposed to 20-50 for alphabets and 40-90 for syllabaries. This creates several difficulties for decipherment:

- computational complexity—our decipherment algorithm runs in time roughly cubic in the number of known sound triples.
- very rare characters—if we only see a character once, the context may not be rich enough for us to guess its sound.
- sparse sound-triple data—the decipherment of a written text is likely to include novel sound triples.

We created spoken language data for Chinese by automatically (if imperfectly) pronouncing Chinese text. We are indebted to Richard Sproat for running our documents through the text-to-speech system at Bell Labs. We created sound-pair frequencies over the resulting set of 1177 distinct syllables, represented in *pinyin* format, suitable for synthesizing speech. We attempted to decipher a written document of 1900 phrases and sentences, containing 2113 distinct characters and no word separators.

Our result was 22% syllable accuracy, after 20 EM iterations. We may compare this to a baseline strategy of guessing the pinyin sound de0 (English "of") for every character, which yields 3.2% accuracy. This shows a considerable improvement, but the speech in not comprehensible. Due to computational limits, we had to (1) eliminate all pinyin pairs that occurred less than five times, and (2) prevent our decoder from proposing any novel pinyin pairs. Because our our goal-standard decipherment contained

many rare sounds and novel pairs, these computational limits severely impaired accuracy.

7 Discussion

We have presented and tested a computational approach to phonetically deciphering written scripts. We cast decipherment as a special kind of text-to-speech conversion in which we have no rules or data that directly connect speech sounds with written characters. We set up general finite-state transducers for turning sounds into writing, and use the EM algorithm to estimate their parameters. The whole process is driven by knowledge about the spoken language, which may include frequency information about sounds, sound sequences, words, grammar, meaning, etc. An interesting result is that decipherment is possible using limited knowledge of the spoken language, e.g., sound-triple frequencies. This is encouraging, because it may provide robustness against language evolution, a fixture of archaeological deciphering.

However, our experiments have been targeted a bit narrowly. We were able to re-use the Spanish decoder on Chinese, but it could not work for Japanese kana. Even our Japanese decoder would fail on an alternative syllabic script for Japanese which employed a single symbol for the sound KAO, instead of separate kana symbols for KA and O. One ambitious line of research would be to examine writing systems in an effort to invent a single, generic "mother of all writing systems," whose specializations include a large fraction of actual ones. To cover Spanish and Japanese, for example, we could set up a scheme in which each sound produces zero or more characters, where the sound is potentially influenced by the two sounds immediately preceding and following it. This gets tricky: the "mother of all" has to be general, but it also has to be narrow enough to support decipherment through automatic training. (Sproat, forthcoming) suggests the class of finite-state transducers as one candidate. This narrows things down significantly from the class of Turing machines, but not far enough for the direct application of known training algorithms.

In the future, we would like to attack ancient scripts. We would start with scripts that have already been roughly deciphered by archaeologists. Computer decipherments could be checked by humans, and published human decipherments could be checked by computer. We would subsequently like to attack ancient scripts that yet to be deciphered. High-speed computers are not very intelligent, but they display a patience that exceeds even the most thorough human linguist.

It will be important to consider text-layout questions when dealing with real scripts. For example, Mayan glyphs may run from top to bottom, right to left, or they may run differently. Furthermore, each glyph contain sub-parts representing up to ten sounds, and these may be organized in a spiral pattern.

Another intriguing possibility is to do language identification at the same time as decipherment. Such identification would need to be driven by online sound sets and spoken corpora that span a very wide range of languages. Whether a document represents a given language could then be estimated quantitatively. In case language identification fails, we may be faced with a completely extinct language. Current computational techniques demonstrate that it is theoretically possible to figure out where nouns, verbs, and adjectives from raw text, but actual translation into English is another matter. Archaeologists have sometimes succeeded in such cases by leveraging bilingual documents and loan words from related languages. Only a truly optimistic cryptanalyst would believe that progress could be made even without these resources; but see (Al-Onaizan and Knight, 1999) for initial results on Arabic-English translation using only monolingual resources.

Finally, we note that the application of source-channel models to the text-to-speech problem is promising. This kind of statistical modeling is prevalent in speech recognition, but ours is one of the few applications in speech synthesis. It may be possible to use uncorrelated streams of speech and text data to learn mappings that go beyond character pronunciation, to pitch, duration, stress, and so on.

References

Y. Al-Onaizan and K. Knight. 1999. A ciphertext-only approach to translation. (In preparation).

L. E. Baum. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3.

J. Chadwick. 1958. *The Decipherment of Linear B.* Cambridge University Press, Cambridge.

M. Coe. 1993. *Breaking the Maya Code.* Thames and Hudson, New York.

J. DeFrancis. 1989. *Visible Speech: The Diverse Oneness of Writing Systems.* University of Hawaii Press, Honolulu.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38.

K. Knight and J. Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4).

F. Pereira and M. Riley. 1997. Speech recognition by composition of weighted finite automata. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing.* MIT Press.

J. Sampson. 1985. *Writing Systems.* Stanford, Stanford.

R. Sproat. forthcoming. *A Computational Theory of Writing Systems.*