An Analysis of Hamptonese Using Hidden Markov Models

Ethan Le
Undergraduate
Department of Computer Science
San Jose State University
San Jose, CA, U.S.A.
Email: lddm933@yahoo.com

Dr. Mark Stamp
Assistant Professor
Department of Computer Science
San Jose State University
San Jose, CA, U.S.A.
Email: stamp@cs.sjsu.edu

Table of Contents

List of Figures and Tables

Tables

Figures

1. Introduction

Throughout history, there have been different types of “secret writing,” new languages, and cipher systems. Mysterious writing and cipher systems persistently torment cryptologists. Unfortunately, the mark of a successful encryption lies in its undecipherable contents, thus, some of the world’s mysterious writings remain a mystery. One such piece of undecipherable writing comes from a recluse by the name of James Hampton. Unfortunately, Hampton’s claim to fame was post-mortem when, in 1964, his landlord discovered his awe-inspiring work of art, “The Throne of the Third Heaven of the Nations’ Millennium General Assembly,” along with Hampton’s undecipherable text, which we have dubbed “Hamptonese.”

The Hamptonese text spans 167 pages. The text contains English words and Bible passages accompanied by strange symbols that are of unknown origin. Refer to Section 3.1, Figure 3, and Table 1 for more information. Next, we will briefly discuss the life and legacy of James Hampton as a way to shed light onto the background of Hamptonese. Then, we will attempt to transcribe and decipher Hamptonese using Hidden Markov Models.

1.1. James Hampton

There is little information about James Hampton’s life or work. Scattered information shows that Hampton was born in South Carolina in 1909. His father was a nomadic Baptist minister and gospel singer. Just before turning 20, Hampton relocated to Washington and made a living as a short order cook. After serving in the Army during World War II, records show that Hampton returned to Washington D.C. and worked as a janitor for the General Services Administration in Washington D.C.

Hampton lived a life of solitude. He never had any close friends and he did not marry. He spent the latter part of his life laboring over his masterpiece, “The Throne of the Third Heaven of the Nations’ Millennium General Assembly.” The Throne has been on display in museums all over the United States, including Minneapolis, New York, Boston, Virginia, and Alabama, and is now a permanent display at the Smithsonian Institute’s National Museum of American Art in Washington D.C. Figures 1 and 2 shows a picture of Hampton’s masterpiece and one of the only surviving pictures of James Hampton himself. The Throne reportedly took Hampton 14 years to construct. It is composed of old junk that included old furniture, burned-out light bulbs, jelly jars, carpet cylinders, desk blotters, cardboard, and foil with glue, tape, tacks, and pins holding the pieces together. The masterpiece “occupies an area of some two hundred square feet and stands three yards in height at its center” (James Hampton, http://www.fredweaver.com/throne/thronebod.html) and is composed of 180 separate objects, each wrapped in silver and gold foil.

> The Throne's many pieces are arranged symmetrically on either side of a main throne chair. Matched pairs of smaller tables and ornaments decorate each side, the objects on the left referring to the New Testament, and, to the right, the Old Testament. The throne chair is crowned with the words "Fear Not," and tacked to a board is the inscription, "Where There Is No Vision the People Perish." Many of

the pieces are tagged, and on the tags, Hampton often refers to himself as "Saint James" (The Miracle of St. James Hampton, http://www.missioncreep.com/tilt/hampton.html).

Hampton told very few, if any, about his "Throne." Construction took place in a small, unheated garage in the rundown section of Washington D.C. In fact, it was not until he succumbed to cancer in a VA hospital in 1964 that Hampton's life work was discovered.

Figure 1. James Hampton and "The Throne of the Third Heaven of the Nations' Millennium General Assembly"
Source: http://www.missioncreep.com/tilt/hampton.html

Figure 2. Pieces of James Hampton's Throne
Source: http://www.fredweaver.com/throne/thronebod.html

Hampton constructed The Throne as a tribute to God, and many believe that it was to serve as the altar and central teaching device when Hampton becomes a minister with his own street-side ministry. Hampton adamantly believed in the Second Coming and his work on the Throne was in preparation for that event. Reportedly, Hampton communicated with God during the construction of "The Throne." Hampton's first vision occurred when he was only 22-years-old.

Many of his visions are recorded on tablets that garnish the Throne. These tablets provide the dates and brief descriptions of Hampton's visions. One message reads, "This is true that the great Moses the giver of the tenth commandment appeared in Washington, D.C. April 11, 1931." Another tablet on the Throne announces, "This is true that Adam the first man God created appeared in Person on January 20, 1949. This was on the day of President Truman's inauguration." Although the Throne exhibits religious connections, "it is worth noting that recognition of the Throne ultimately came from the artistic community, not the religious community" (The Miracle of St. James Hampton, http://www.missioncreep.com/tilt/hampton.html).

Along with the Throne, Hampton's legacy lives on in several notebooks filled with strange writings discovered among the remnants of his lonely garage. These writings, or Hamptonese, still baffle cryptologists. Because little is known about Hampton's life and the "visions" that inspired the construction of the Throne, many believe that Hamptonese is just the senseless writings of an eccentric man who lived a life of poverty, loneliness and obscurity. If Hamptonese carries any meaning, we would like to decipher that meaning. If Hamptonese is just gibberish, we would like to show that is the case.

2. Purpose

Hamptonese is a writing system that has remained a mystery for 40 years. To this day, there has been no serious research done to decipher the meaning of Hamptonese. Previous attempts to decipher Hamptonese have been of limited scope and yielded little convincing evidence. One such research is available at Hamptonese Statistics (Hamptonese Statistics, http://www.geocities.com/ctesibos/hampton/hamptonese.html). Thus, there is a need for further research to decipher Hampton's writing. In our research, we wish to determine whether the writing is simply gibberish or a simple substitution of English or other language. Our research concentrated on the last 103 pages of Hampton's text, which contains pure Hamptonese. We analyzed the Hamptonese using Hidden Markov Models in hopes of undeniably determining whether Hamptonese was created using a simple substitution of English letters or sounds (or other languages). To consider other perspectives, we also analyzed the actual Hamptonese pages to determine if there are certain patterns or hints that might help us decipher Hamptonese.

3. What is Hamptonese?

Hamptonese is the writing left behind by James Hampton when he died in 1964. It was discovered among the remnants of Hampton's belongings in a garage in Washington D.C. His writing filled several notebooks and spans 167 pages. These symbols are written without any signs of punctuation or paragraphs. The text is interspersed with a few sporadic English words and references to Bible passages. A partial page of Hamptonese is seen in Figure 3. Scanned pages of Hamptonese are available for download at http://www.cs.sjsu.edu/faculty/stamp/Hampton/pages.html.

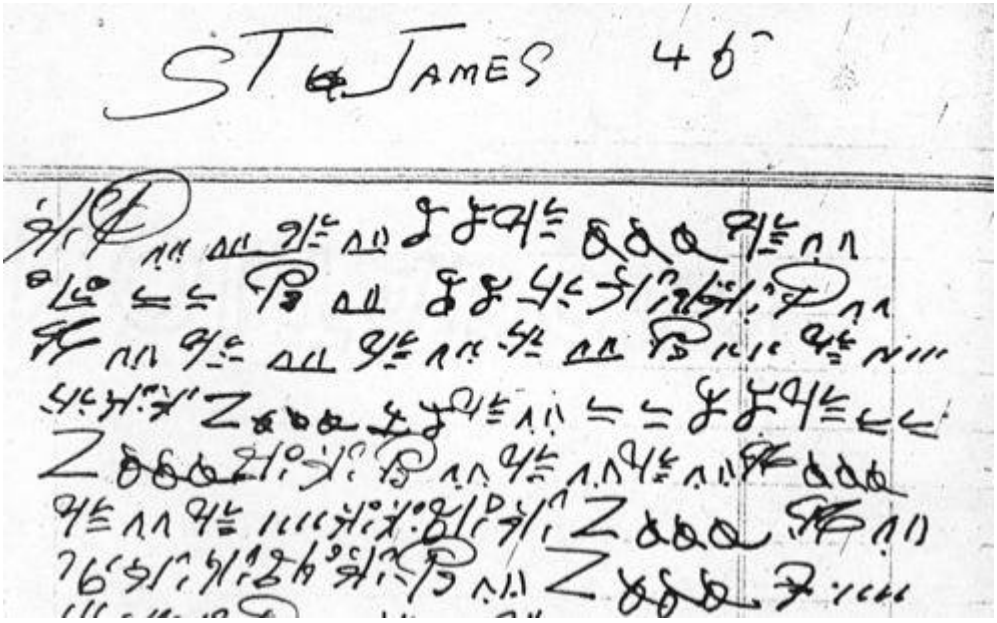


Figure 3. Partial page of Hamptonese

Source: http://www.cs.sjsu.edu/faculty/stamp/Hampton/hampton.html

3.1. Description of Hamptonese text

The first 62 pages of the text include pictures, English text, and Bible passages. Hampton divided the first 48 pages into 10 sections, each with distinguishable recurring patterns. Each section leads off with a page that contains a drawing (at the center of the page) of a "tombstone" with two columns of Roman numerals. The left side contains the Roman numerals one through five written vertically, and the right side contains Roman numerals six through ten, also written vertically. On either side of the tombstone drawing are nine lines of Hamptonese text. At the top is a line of Hamptonese that varies from section to section, so it is reasonable to assume that this might be a title line or some kind of numbering system. At the bottom of the page is a line of Hamptonese followed by a recurring message: "THE OLD AND THE NEW COVENANT RECORDED BY ST JAMES."

Following each such page is a page that has a Hamptonese symbol centered at the top. Directly below the symbol is a line of text that reads, "ST James RECEIVED Jesus BY MOSES." Then the page is divided into 2 columns. The left column contains Hamptonese while the right column contains 25 lines of English text, in particular, a

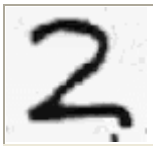
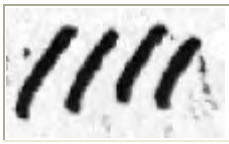
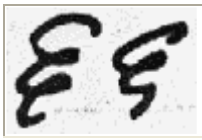
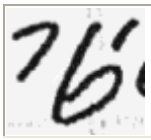
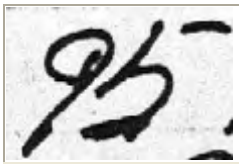
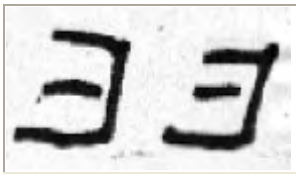
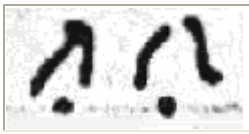
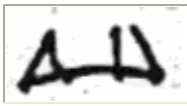
passage from the Bible. The passage comes from Exodus 20:1 to Exodus 20:7 (BibleGateway, http://bible.gospelcom.net/). Only 19 out of the 25 lines begin with Roman numerals. Following this type of page, Hampton occasionally inserts pages of Hamptonese or various Bible passage. Pages 49 through 62 of the text are devoted to what Hampton calls “7 Dispensations.” These pages feature the Seven Dispensations for Moses, Elijah, the Virgin Mary, and Adam, each of which is dated. Interestingly, the date of some of these Dispensations coincides with dates of Hampton’s “visions” as inscribed on the Throne.

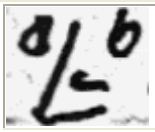
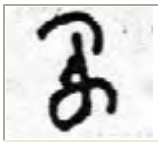
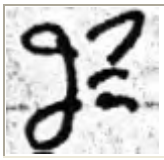
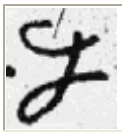
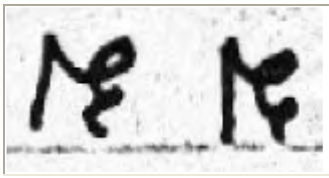
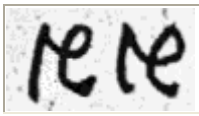
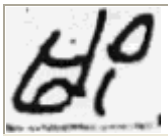
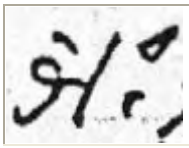
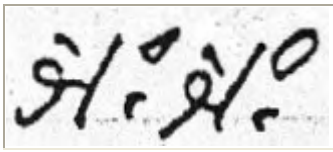
Following Hampton’s Dispensations are the 103 pages of pure Hamptonese text. These pages run from page 64 to page 167 (103 pages, though three pages are duplicates so there are exactly 100 distinct pages). There is no sign of punctuation or paragraphs. Each page of the text is headed by the words “St James” followed by a single or double-digit number at the top. Scattered among the pages are a few English words and letter combinations. The English word list include “Jesus,” “Christ,” James,” “NRNR,” and “Revelation.” Notably, only the word Jesus is written as “Jesus,” the rest of the words and letters are all capitalized. Every page contains exactly 25 lines of text. However, each line does not contain the same number of symbols. Moreover, although most of the text seems to be written in horizontal lines, there are, on some of the pages, distinct separations of text into lines, columns, and shades of black and gray. Section 11.3 examines these patterns in more details.

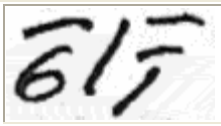
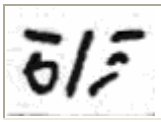
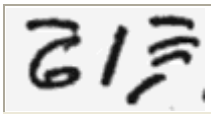
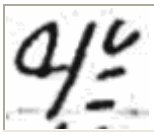
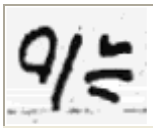
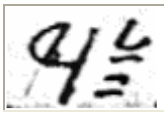
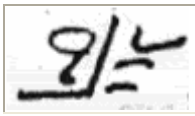
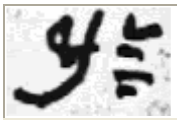
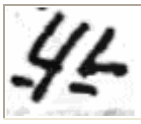
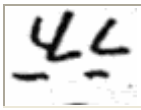
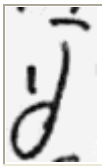
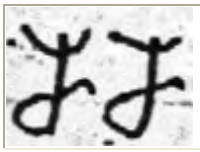
3.2. Transcription

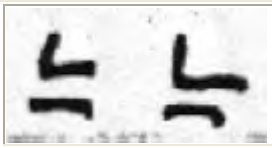
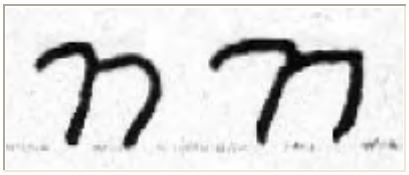
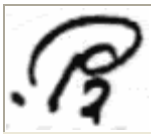
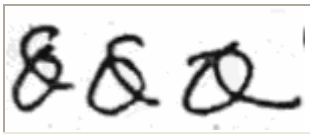
In order to perform the analysis of the Hamptonese text, the text had to be transcribed into discernable symbols. To do this, we assigned each of 56 distinct Hamptonese symbols a corresponding arbitrary English grouping of letter(s) and/or number(s). For example, ‘14’ was used to denote the Hamptonese symbol that consists of four vertical markings. Table 1 contains the Hamptonese symbols (on the right) alongside its English transcription (on the left). It is important to note that Hampton left no guidelines for how the symbols are to be interpreted. Therefore, when we state that there are 56 different symbols, it is based on our own interpretation of the groupings of the symbols. This is just our way of organizing the Hamptonese symbols.

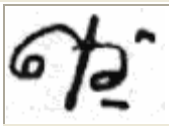
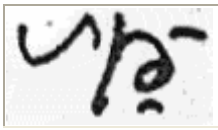
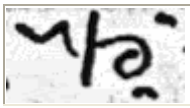
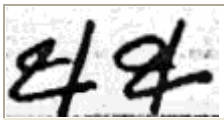
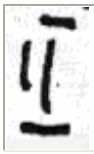
Table 1: Transcribed Symbols Alongside Hamptonese

2	
14	
ee	
76	
95	
96	
dc	
EE	
vv	
M	
F	

N	
G	
g (variant)	
G7	
Y	
O1	
GG	
HH	
Gi	
Ki	
Ki Ki	

D3	
d3 (variant)	
D4	
Y3	
Y3 (variant)	
Y4	
qL3	
qL4	
4L	
uL	
J1	
J2	
JJ	

LL	
LLL	
nun	
P	
P (variant)	
PL	
P2	
O3	
Q3	
S	
3	
T	

A	
A-	
A-(variant)	
44	
I	
010	

Source: http://www.cs.sjsu.edu/faculty/stamp/Hampton/hampton.html

While transcribing Hamptonese, there were subtle differences in the symbols that resulted in an expansion of possible Hamptonese symbols. For example, markings missing and added to the Y3 symbol resulted in additional symbols, which we denoted as Y0, Y1, and Y2. Thus, the list of translated Hamptonese symbols expanded to 70 symbols; though, undoubtedly many of these are simply Hampton's "typos." Table 2 displays the expanded list of observed Hamptonese symbols. Furthermore, it is noteworthy that the pages of Hamptonese used in our research were photocopies of microfilm. In addition, some of the pages contain unclear symbols, missing edges, water damage, and printing damage. These quality variations did affect our transcription process. Due to the inconsistency of quality, there were symbols that were undecipherable or unknown and could not be grouped with our list of Hamptonese symbols. The unclear (or unsure) symbols are denoted by a preceding '#' sign, and the unknown symbols are denoted by '*'. Finally, English words interspersed in Hamptonese were transcribed by enclosing them within brackets. See Table 3 for an example of the transcription process.

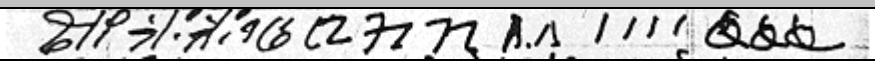
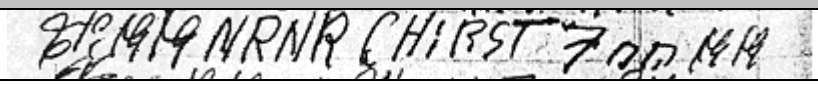
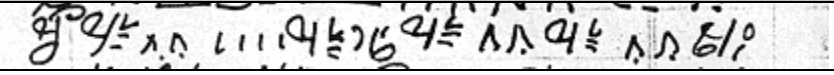
After determining our set of transcription symbols, all 100 pages of Hamptonese were transcribed and input into a text document. Our computer programs later used this document to analyze Hamptonese.

Table 2. Expanded List of Observed Hamptonese Symbols

010	13	14	15	2	3	4	44	76	95
96	4L	d2	d3	d4	dc	e	ee	g	g7
n	nn	o3	q3	qL	qL0	qL1	qL2	qL3	qL4
uL	v	vv	Y	A	A-	E	EE	F	Gi
GG	HH	I	J	J1	J2	JJ	Ki	KiKi	L4
L	LL	LLL	M	N	O1	P	PL	P1	P2
P3	P4	S	T	Y0	Y1	Y2	Y3	Y4	Y5

Source: Le and Stamp, 2003

Table 3: Transcription Process

Hamptonese	
Transcribed	Gi Ki Ki 76 * nn vv 14 q3
Hamptonese	
Transcribed	Gi GG [NRNR] [Christ] F vv GG
Hamptonese	
Transcribed	#g Y3 vv 14 Y4 Y3 vv Y3 vv Gi

Source: Le and Stamp, 2003

3.3. Frequency Counts

To observe the relative frequencies of the Hamptonese symbols, we coded a simple count program. The program's purpose was to count the occurrences of valid Hamptonese symbols using the expanded list of Hamptonese symbols (see Table 2).

The count program operated with two settings that allowed users to either include or exclude the undecipherable (denoted by '*') and unclear (denoted by a preceding '#') Hamptonese symbols. The program proved effective in detecting typos, which were immediately corrected. Furthermore, an interesting collateral benefit of the count program was its detection of certain symbols that did not occur in the transcription at all. These symbols were subsequently eliminated as valid Hamptonese symbols. We used the results of the frequency counts to group symbols based on their relative frequency. The process of grouping is explained in more detail in Section 8. See Table 4 in Appendix B for the results of the frequency count program. Table 5 in Appendix B shows the results of the frequency count after Hamptonese symbols were grouped. For information about grouping of Hamptonese symbols, see Section 8.2).

The results revealed that, excluding the 228 undecipherable symbols, there were 29297 Hamptonese symbols, with 229 of those marked as unsure translations. This information was used to determine the parameters of the Hamptonese analysis using Hidden Markov Models. After performing the tedious, yet essential, transcriptions of the text, and recording the occurrences of each symbol, Hidden Markov Models were applied to test the possibility that Hamptonese is a simple substitution cipher.

However, before discussing our HMM, we need to discuss the basic concepts of HMMs as well as the properties of HMMs that are applicable to our project. Here, we only present a high-level overview of HMMs. For more information, see "A Revealing Introduction to Hidden Markov Models" (Stamp, 2003).

4. Hidden Markov Models (HMMs)

Suppose that the Search for Extraterrestrial Intelligence (SETI) Institute, in hopes of making contact with Martians, sent a transcription of the Gettysburg Address to the surface of Mars. Suppose further that Martians do exist and they received our message. What would the Martians make of this strange speech? How could they determine what it means? How can they determine whether it is a language, a drawing, or something else? If Martians knew the Hidden Markov process, they could run the Gettysburg Address through the process and be much closer to deciphering the strange message.

Hidden Markov Models are a generalization of a Markov Model. The Markov process simply relates the probability of an event happening from one time to the next. The Hidden Markov Model, however, supposes that you do not know the actual events taking place. These events are "hidden" behind a curtain. The "observations" happening in front of the curtain are all that we know. Using the Hidden Markov Model, we can derive some information about what is going on behind the curtain. Essentially, HMMs provide probabilistic information about the underlying state of a model, given a set of observations of the system.

In applications to undecipherable text such as Hamptonese, Hidden Markov Models allow the user to decode substitution patterns in text that the user has no prior knowledge. All that the user needs to do is to transcribe the text into symbols or characters that he or she recognizes. Once the Markov Model processes are completed, the "hidden" parts of the Markov Model are revealed (at least probabilistically). In other words, the text is the set of observations and the underlying (and unknown) language is the hidden part of the HMM. The results of the HMMs will show a separation of the characters into different states. An English text example is explained in detail in Section 5. For further information on HMM, refer to "A Revealing Introduction to Hidden Markov Models" (Stamp, 2003), "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" (Rabiner, 1989), and "Hidden Markov Models for English" (R.L. Cave and L.P. Neuwirth, 1980).

4.1. Hidden Markov Models Applications

The intricacies of Hidden Markov Models can be difficult to grasp and understand. However, there are copious applications of Hidden Markov Models. The best-known application of HMMs is in the area of speech recognition.

4.1.1. HMM In Speech Recognition Algorithms

The best-known application of HMMs is in speech recognition algorithms. In this case, an HMM model is built for each possible word that can be observed. Then the HMMs are trained over repetitions of the occurrences of these observations to refine itself into a model that best fits the observation sequence.

For example, suppose that a speech recognition program is used in a cell phone to allow users to dial a number by simply speaking a name. The cell phone will use an HMM to model the likelihood that a spoken name corresponds to a specific number. In setting up the correlation between the spoken name and the number, the user would first have to repeat the name several times into the receiver. This is because at each repetition of the sound, an HMM model is built and trained to the variations of that particular sound.

As another example, consider a typical PC program that allows users to speak into a microphone and the program will type the spoken words. Now suppose that the program uses HMMs to find the word that best matches a spoken word. In this case, before the program is released, the company would have to train the HMM model repeatedly. In all likelihood, different people would speak the letter 'a' into the microphone, at each repetition of the letter, the HMM would find the best model that fits the spoken word. After hundreds of repetitions, the HMM will have found the optimal model for the sequence of repetitious 'a'. This process would yield an HMM that specifically corresponds to the way the letter 'a' can be spoken. In a speech program, this process would have to be run hundreds of thousands of times, each generating its own HMM that specifically corresponds to a particular letter or word. Now, whenever a user speaks a word or a sentence, the program would search for the best model that fits the given observation.

For more information about the intricacies of speech recognition using HMM, refer to “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition” (Rabiner, 1989).

4.1.2. Music-Information Retrieval and HMMs

HMMs have also proved useful in developing music-information retrieval (MIR) system.

In work by Birmingham et al., a team developed an MIR system using the forward pass of an HMM to eliminate errors in their MIR system. The researchers asked users to sing a certain portion of three predetermined songs. These segments were recorded and input as observations. Then, users repeated the process for three other songs of their choice, picked from a database of 277 pieces of music. These again were recorded. Due to the inherent errors of tone, pitch, voice modulations, and the accuracy of the sung segments, the researchers chose to use HMM as a way to reduce the errors.

> As the error sources are non-deterministic (we cannot accurately predict what error a given user will make on a given piece of music), we have chosen to use a stochastic representation of music in our system. In particular, we use hidden Markov models (HMM) to represent both queries and themes. Other researchers have used imprecise string matching to deal with errors, where the query and the target (either a theme or a full piece) are treated as strings with "noise". We believe that an HMM approach is more flexible than noisy strings, as we can explicitly model different error processes in the most suitable way. (Birmingham et al, 2002).

In using the HMM, the researchers were able to retrieve the particular songs or music pieces that were “queried,” with few errors.

> Hidden Markov models are an excellent tool for modeling music queries. The results of our experiments with this "first-step" implementation indicate both the promise of these techniques and the need for further refinement. Refinements to the hidden model topology and of the observation model will allow us to model a broader range of query behavior, and improve the performance of the system. (Birmingham et al., 2002).

For more information about similar MIR systems research, refer to “Folk Music Classification Using Hidden Markov Models” (Chai and Vercoe, 2001), and “Automatic Segmentation for Music Classification using Competitive Hidden Markov Models” (Battle and Cano, 2000).

MIR systems are not only of interest in research, these concepts have been applied to software that is currently on the market. For example, iTunes, a popular music download, storage, and player system for Macintosh computers has recently utilized the ideas of HMMs for MIR systems to allow its users to simply hum a tune and obtain a list of all the songs in the user’s library that have similar tunes. Therefore, when an iTunes user has a melody stuck in his head and the song’s title on the tip of their tongue, iTunes will allow them to retrieve the song quickly. This innovative concept was

presented in the article “Music Information Retrieval Systems” (Brimingham, Meek, O’Malley, Pardo and Shifrin, Dr. Dobb’s Journal, September 2003).

4.1.3. English Alphabet Analysis Using HMMs

As mentioned above, HMMs can be used to classify language characters (which is our goal). Using HMMs, we can process any passage of English text (if the passage contains enough characters for the model to be accurate). In particular, after HMM analysis of the English text, we might expect to see vowels and consonants of the English alphabet belonging to different categories, or states. The implementation of this approach is presented in Section 5.

5. English Text Analysis Using Hidden Markov Models

A program was written to analyze English text using the Hidden Markov process. We considered two hidden states.

We coded the Hidden Markov Model in both C and Java. In either case, the model had 2 hidden states and 27 different observation symbols. These observation symbols included the 26 letters of the English alphabet plus the space character. All upper-case letters were converted to lower-case and all punctuation, symbols, etc., were discarded. The program also allowed the user to specify the input text document, the seed number to be used to generate random numbers during initialization, the starting position in each line, how many characters will be ignored, how many characters will be considered, and how many iterations of the Markov Model the program will run. Figure 4 elaborates on the command line parameters needed to run the program in C language. Unlike the C program, the Java program did not run with command line arguments, instead, all of the command line argument parameters were read in from a file.

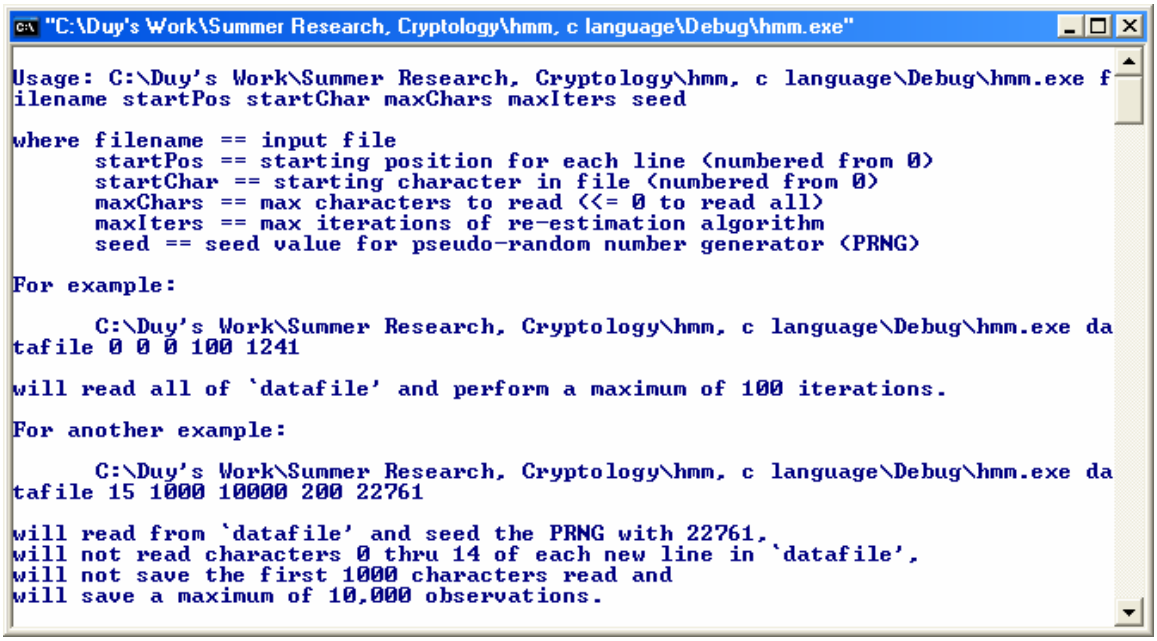


Figure 4. Explanation of Hidden Markov Model Program in C

Source: Le and Stamp, 2003

Once the programs were coded, we needed to find a document of English text to have the program process. In our case, we decided to use the Brown Corpus document (Natural Language Computing, http://www.cs.toronto.edu/~gpenn/csc401/a1res.html) because it contained a sufficient number of English characters needed for the program to run efficiently. The format of the Brown Corpus text helped determine the values of the parameters. Figure 5 shows a few lines of the Brown Corpus document. When running the HMM, we set our parameters to start reading at the 15th position, ignore nothing, run for 200 iterations with seed 22761, and save a maximum of 10000 observations. Starting at the 15th position means that the leading character and numbers of each line would not be considered (i.e. "A01 0010 1" or "A01 0020 9" would be excluded in the observations). Once the model has performed all 200 iterations, the results will show the probability of each letter belonging to either one of the two states. Table 6 shows an excerpt of the English HMM results, for the full list of results see Table 6 in Appendix B.

```
A01 0010  1      The Fulton County Grand Jury said Friday an investigation
A01 0020  1    of Atlanta's recent primary election produced "no evidence"
A01 0020  9    that any irregularities took place.
```

Figure 5: Brown Corpus example
Source: http://www.cs.toronto.edu/~gpenn/csc401/a1res.html

Table 6. Excerpt of English HMM results

T = 10000 N = 2 M = 27 iterations = 200

final pi = 1.0000000 0.0000000 , sum = 1.0

final A =
0.2755622 0.7244378 , sum = 0.9999999999999982
0.7294855 0.2705145 , sum = 1.000000000000003

final B^T =

A	0.0044447	0.1306242
B	0.0241154	0.0000000
C	0.0522168	0.0000000
D	0.0714247	0.0003260
E	0.0000000	0.2105809
.	.	.
.	.	.
Sum	0.9999999999999946	1.0000000000000044

Log[P(observation | lambda)] = -27511.531341430866

Source: Le and Stamp, 2003

From the results (Tables 6 and 7), it is clear that the vowels belong to one state while the consonants belong to another. While this is not surprising, it is important to note that the HMM contains no explicit information about consonants or vowels. This information is deduced by the HMM due solely to the statistical information in English text. This

implies that if Hamptonese is a simple substitution for English letters, an HMM will separate the Hamptonese symbols into those that correspond to vowels and those that correspond to consonants. It is worth noting that it is possible to get sensible results with more than two hidden states (R.L. Cave and L.P. Neuwirth, 1980). The results of HMM analysis of English letters under three, four, and five hidden states are available online at http://www.cs.sjsu.edu/faculty/stamp/Hampton/hampton.html. These results show that English letters show distinct separations into different states for three, four, and five hidden states.

Furthermore, both the C and Java programs gave the same overall expected results with slight numerical differences in the initial and final matrices due to different random number generators. However, these differences were less then six decimal places, therefore, negligible. Nevertheless, in order to insure that the actual model work correctly after it was translated from C to Java, two trials of the Java HMM programs were ran. The first trial utilized the random number generator of the Java library and the second used the random numbers of the matrices generated from the C program. The second trial yielded the same final values as the C program, thus, proving that the translation was correct. This is important because testing of the Hamptonese text using Hidden Markov Model would only use the Java program.

6. Modeling The Hamptonese HMM

In our application of HMMs, we are given an observation sequence (i.e. Hamptonese text) along with N (the number of states) and M (the number of observations) and we wish to find the model that maximizes the probability of the observed sequence. Again, for more information on HMMs and their implementation, see “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition” (Rabiner, 1989), or “A Revealing Introduction to Hidden Markov Models” (Stamp, 2003).

In our research, the transcribed Hamptonese text is the observation sequence. Additionally, the number of hidden states will vary from N equal 2 to 5, the number of observation symbols will fluctuate between M equal 47 and 66 (this fluctuation is explained in Section 8), and the matrices that define the model (namely A, B, and π) are initialized by 3 different arbitrary seed values (see Section 7.2).

7. Hamptonese Analysis

The first step in analyzing Hamptonese was to determine the different ways that the text can be interpreted. To this end, we applied our HMMs to the Hamptonese with many reading techniques and interpretations.

7.1. Reading Techniques

Forward Reading of Hamptonese

As mentioned above, the Hidden Markov Models for Hamptonese would operate under three different seed values and varying iterations. First, 29297 transcribed Hamptonese symbols were read in and considered as the set of observations. The observations were read line by line from left to right.

Backwards Reading of Hamptonese
To anticipate other reading methods for Hamptonese, the Hamptonese text was processed by the HMMs reading the entire text backwards. This means that the first observation recorded by the model would be the last Hamptonese symbol of the transcribed text, and the last observation recorded by the model would be the first Hamptonese symbol of the transcribed text.

Backwards By Line Reading of Hamptonese
The Hamptonese text was also run through the HMMs backwards line by line. This means that the entire transcribed text was read in from front to back, however, the observations (symbols) of each line were recorded from the end of the line to the beginning of the line. See Figure 6 for an example of this process.

Transcribed passage	Recorded in HMM as…
Ki P vv Ki dc GG M g * qL4 14 GG vv g 2 o3 Ki qL3 M g * J1 N S q3 44 vv g vv qL3 M M Y3 Y3 vv JJ	vv GG 14 qL4 * g M GG dc Ki vv P Ki q3 S N J1 * g M qL3 Ki o3 2 g JJ vv Y3 Y3 M M qL3 vv g vv 44

Figure 6. Example of Reading Backwards By Line
Source: Le and Stamp, 2003

7.2. HMM Parameters

As mentioned above, the Hidden Markov Model program depended on certain settings (or parameters) to operate properly. The program allowed the user to specify the input text document, the seed number to be used to generate random numbers during initialization, the starting position in each line, how many characters will be ignored, how many characters will be considered, and how many iterations of the Markov Model the program will run. In its application to Hamptonese, the text document contained the transcribed Hamptonese, the starting position was set at the beginning of each line with no symbols ignored and 30000 Hamptonese symbols were considered. Although a maximum number of symbols to be considered were set at 30000, there were only a maximum of 29297 Hamptonese symbols transcribed and available. This limit was set to ensure that all valid Hamptonese symbols were processed. These settings were permanent throughout our analysis; however, other settings varied.

Instead of the HMMs operating on one seed value, as was the case in our English text example, the Hamptonese HMMs were run 3 times, each with a different seed. The three seeds were 22761, 123456, and 333333. These three seeds were picked randomly and were used to define different initial starting points for our model. This is a way of double-checking our results for consistency. The seed values themselves are arbitrary. Their main purpose is to establish a starting point for our model and determine how long the model would run. It is also arbitrary because if the unknown (i.e. Hamptonese text) was developed by simple substitutions, then regardless of the seed values, the results should show similar separation for each observation.

Running all three models (with their respective seed number) with the same number of iterations might yield inconsistent results. For example, if we were to run each of the models with 200 iterations, each model's change in slope would be different and some might not reach their maximum. For instance, using seed value 333333, the initial values might start extremely close to the maximum, therefore, with 200 iterations, the maximum would have been reached and the final values are as desired. On the other hand, another seed, for example, 22761, may start extremely far from the maximum; therefore, with 200 iterations, the model would not yield the optimal final values. Due to these variations, the number of iterations was changed to be dependent on the slope changes of the model. The program was set to run a minimum of 200 iterations and stop when the slope change differs by more than six significant decimal digits.

The final change in our HMMs was to consider only 66 possible Hamptonese symbols, instead of the original 70. This change was a direct result of the symbol count (see Tables 4 and 5 in Appendix B). As seen in the count results, 4 symbols did not appear at all, therefore, should not have been included to begin with. Aside from these necessary changes, the Hidden Markov Model mirrored that of the one used for our English text example (see Section 5).

8. Hamptonese HMM Results

8.1. Non-Grouped

We first ran the Hamptonese transcribed text using our HMMs three times (each with a different seed value) with each symbol representing an observation. All 66 possible Hamptonese symbols were considered valid observations and the observations were recorded in a "forward reading" manner. Note that our original expansion of Hamptonese was up to 70 symbols, however, the model only use 66 possible Hamptonese symbols. This is because four Hamptonese symbols did not occur in the Hamptonese text of interest. From the results (see Table 8), it was clear that there is no distinct separation of the symbols into either of the two hidden states. In our research, separation of a symbol into a distinct state can only occur if the numerical value of that state compared to the other state's numerical value is greater than or equal to two significant digits (i.e. 0.001).

The general belief with HMMs is that if separation is not distinct with a two states model, the separations of the observations will not be improved by expanding the number of possible states. However, since Hamptonese has an unknown origin, the separation of its observations might exist in any number of states models. Therefore, we expanded our list of possible states to three, four and five and processed the observations based on three different seeds for each of the states in the "forward reading" manner. Again, this did not show any significant separation. Table 8 features a list of the separation in each state and Table 11 shows the running time for all the HMMs.

Table 8. Hamptonese Results with 66 Symbols, Reading Forward

	Seed = 22761	
States	*2*	13, 2, d2, d3, d4, e, n, o3, qL, qL0, qL1, qL2, qL3, qL4, v, vv, F, J, P, P1, P2, P3, P4, Y0, Y3, Y4, Y5
	3	96, e, n, o3, qL, qL0, qL1, qL3, v, JJ, Ki, N, P, P1, P4, S, Y0, Y5
	4	13, 15, e, qL0, v, Ki, P4, S, Y0, Y1, Y5
	5	010, 13, qL0, v, Ki, P4, S, Y0, Y1, Y5
	Seed = 123456	
States	*2*	13, 2, d2, d3, d4, e, n, o3, qL, qL0, qL1, qL2, qL3, qL4, v, vv, F, J, P, P1, P2, P3, P4, S, Y0, Y3, Y4, Y5
	3	010, 13, e, o3, qL0, v, vv, J, Ki, P4, Y0, Y1, Y3, Y5
	4	010, 13, e, n, o3, qL0, v, vv, Ki, Y0, Y1, Y5
	5	010, 13, e, o3, qL0, v, P4, S, Y0, Y1, Y5
	Seed = 333333	
States	*2*	010, 13, 96, d3, d4, e, n, o3, q3, qL, qL0, qL3, qL4, v, vv, I, J2, Ki, L, P4, S, Y0, Y1, Y2, Y3, Y5
	3	010, 13, e, o3, qL0, v, Ki, P4, S, Y0, Y1, Y5
	4	13, 96, d4, e, n, o3, q3, qL0, v, vv, Ki, P, P3, P4, Y0, Y1, Y3, Y5
	5	13, e, n, o3, qL0, v, J2, Ki, P4, S, Y0, Y5

Source: Le and Stamp, 2003

8.2. Grouped

Realizing that our HMMs did not show any distinct separation (thus, a failed result), we had to consider alternate ways to proceed. The next step was to consider possible groupings of Hamptonese symbols. As mention above (see Sections 3.2 and 3.3), the transcription and counting process revealed that certain Hamptonese symbols were underrepresented. Thus, these underrepresented symbols can be grouped with similar symbols. For example, since Y0, Y1, Y2, and Y5 appeared infrequently in the text and Y3 occurs quite often, Y0, Y1, Y2, and Y5 were grouped together with Y3. For a complete list of all the groupings, please refer to Table 9**.** For HMMs, the smaller the list of possible observations, the less data that will be required. Thus by trimming our list of valid Hamptonese symbols, we will obtain results that are more accurate. In application to Hamptonese, this works out quite nicely since the expansion of our valid Hamptonese symbols was a direct result of slight variations in the text. Furthermore, redundancy calculations (see Section 10.2) show that groupings of Hamptonese symbol are justified. Therefore, it is valid to group certain symbols if they resemble each other. The decision

of which symbol would group with which is subjective, although we did follow certain rules:

1) Pick out, from the list of all occurrences of a symbol, the low counts.
2) The Hamptonese symbols with low counts are then compared with similar Hamptonese symbols. If the symbols appear to be a slight variation of each other, the low count Hamptonese symbol are considered as part of the higher count Hamptonese symbol.

Table 9. Grouping of Hamptonese

Hamptonese	Considered as/Grouped with
13, 15	14
d2	d3
e	ee
n	nn
qL, qL0, qL1, qL2	qL3
v	vv
E	EE
J	J1
L	LL
P3, P4	P2
Y0, Y1, Y2, Y5	Y3

Source: Le and Stamp, 2003

Once the symbols were grouped, our list of possible Hamptonese symbols shrank from 66 to 47. Thus, we reran the transcribed Hamptonese HMMs with a smaller list of valid Hamptonese symbols under the different variations of seeds and reading techniques.

The HMMs results for Hamptonese with only 47 possible Hamptonese symbols (under the “forward reading” technique) were slightly better then that of our previous runs. The two states model with seeds 22761 and 123456 showed the same 15 separated symbols. Not all symbols separated, as we had hoped. However, the 15 symbols that showed separation all had the same values. Unfortunately, inconsistency occurs with the model using seed 333333, which also contained 15 separated symbols, but only eight of those matched the separations seen using seeds 22761 and 123456. Table 10 summarizes these results.

Because the seed values are arbitrary and we expect all separations (regardless of seed values) to be the same for a simple substitution cipher system, the inconsistencies of our separations suggests that perhaps Hamptonese was not created by a simple substitution cipher. The HMM was also run with for three, four, and five states with the reduced observed symbols list. Again, there was no clear consistent separation of the symbols.

Because inconsistencies were prevalent in both the expanded and reduced list of valid Hamptonese symbols with our HMMs, we had to consider other ways the text could be interpreted. We considered that maybe Hampton meant for the text to be read in a

different way. Perhaps, the text was meant to be read backwards or backwards by line. To this end, the HMMs were run for both cases where observations were read and recorded backwards and read and recorded backwards by lines. For each case, we utilized all three seed values. Unfortunately, the results of these runs showed the same inconsistencies. Seeds 22761 and 123456 had the exact same separation of Hamptonese symbols with same values. In addition, seed 333333 had similar separations but not all matched that of the other two runs. Table 10 summarizes the results of the two states models and Table 11 features the running times for all the HMMs.

These models (with their respective reading techniques and seed values) were also subjected to HMMs of three, four, and five states. Unfortunately, yet again, there was no consistent separation of all of the symbols. However, with four hidden states and reading the observations “backwards by line” and “backwards by whole,” there was consistent separation of four Hamptonese symbols. Furthermore, with 5 hidden states, reading “backwards by line” showed 3 distinct separation of Hamptonese symbols and reading “backwards as a whole” showed 2 distinct separation. This suggests that Hamptonese was meant to be read backwards by line or backwards by whole and contains four or five different states. However, these separations did not match separation observed with the English text example (subjected under three, four and five hidden states), so Hamptonese was not created by simple substitution of English letters. Table 20 in Appendix B shows all separation results for the Hamptonese HMMs. More exclusive results are available online at http://www.cs.sjsu.edu/faculty/stamp/Hampton/Hampton.html.

Due to the recurring inconsistencies obtained from the HMMs, it appears that Hamptonese was not created by a simple substitution of English letters.

Table 10. Hamptonese Results with 47 Symbols and 2 States

Forward Reading		
		Separation in…
Seed	22761	2, 4, d3, d4, o3, qL3, qL4, vv, F, P, P1, P2, S, Y3, Y4
	123456	2, 3, d3, d4, o3, qL3, qL4, vv, F, P, P1, P2, S, Y3, Y4
	333333	010, 96, d3, d4, o3, q3, qL3, qL4, vv, I, J2, Ki, P, S, Y3
Backwards as a Whole Reading		
		Separation in…
Seed	22761	2, 4, d3, d4, o3, qL3, qL4, vv, F, P, P1, P2, S, Y3, Y4
	123456	2, 4, d3, d4, o3, qL3, qL4, vv, F, P, P1, P2, S, Y3, Y4
	333333	010, 2, 96, d3, d4, o3, q3, qL3, qL4, vv, I ,J2, Ki, P, S, Y3
Backwards By Line Reading		
		Separation in…
Seed	22761	2, d3, d4, o3, qL3, qL4, vv, F, P, P1, P2, S, Y3, Y4
	123456	2, d3, d4, o3, qL3, qL4, vv, F, P, P1, P2, S, Y3, Y4
	333333	2, 3, 96, d3, d4, o3, q3, qL3, qL4, vv, I, J2, Ki, P, S, Y3

Source: Le and Stamp, 2003

Table 11. Running Time of the HMMs

Forward Reading with 66 Symbols			
	Seed		
States	22761	123456	333333
2	3m	3m	5m
3	30m	23m	20m
4	42m	1h 25m	1h 30m
5	3h 6m	1h 8m	1h
Forward Reading with 47 Symbols			
	Seed		
States	22761	123456	333333
2	2m	3m	4m
3	8m	38m	45m
4	25m	3h 11m	1h 40m
5	27m	3h 39m	2h 59m
Back By Whole Reading with 47 Symbols			
	Seed		
States	22761	123456	333333
2	3m	2m	6m
3	46m	45m	45m
4	46m	45m	45m
5	3h 3m	2h 15m	1h
Back By Line Reading with 47 Symbols			
	Seed		
States	22761	123456	333333
2	3m	4m	10m
3	5m	52m	11m
4	47m	44m	50m
5	4h 24m	4h 43m	1h 21m

h = hour
m = minute
* NOTE: running times are approximated to the nearest minute

Source: Le and Stamp, 2003

9. English Phonemes

Because Hamptonese and English text HMMs were not similar, thus, proving that Hamptonese was not created by a simple substitution of English letters, we had to find alternative interpretation of Hamptonese.

Logically, we considered whether Hamptonese was a substitution for English sounds, or phonemes. To test this notion, we performed HMM analysis on English phonemes using the previous HMM program (see Section 5 and 7.2) and the Brown Corpus document (Natural Language Computing, http://www.cs.toronto.edu/~gpenn/csc401/a1res.html). The model had 2 to 5 hidden states and 39 or 72 observations. The observation symbols were that of English pronunciation obtained from Carnegie Mellon University (The CMU Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict). For 39 observations, we only considered the base phonemes (see Table 12). For 72 observations, we considered the base phonemes and additional pronunciations that included lexical stress (see Table 12). The program read in the Brown Corpus document, converted all letters to upper case and systematically converted each English word to their respective English pronunciations (see Table 13 for an example). It is worth noting that only words that are in the Carnegie Mellon pronunciation dictionary were considered as valid observations. Furthermore, words with attached punctuation, symbols, etc. (i.e. “Hi) were discarded. Similar to the English text and Hamptonese HMM, the program allowed users to specify certain documents and settings. In this case, the user can specify the pronunciation dictionary file name, the input text document, the maximum number of observations, the minimum number of iterations, and the seed number to be used to generate random numbers during initialization.

Table 12. English Phonemes List

	Phonemes
Base Phonemes	AA, AE, AH, AO, AW, AY, B, CH, D, DH, EH, ER, EY, F, G, HH, IH, IY, JH, K, L, M, N, NG, OW, OY, P, R, S, SH, T, TH, UH, UW, V, W, Y, Z, ZH
Phonemes with Lexical Stress	AA0, AA1, AA2, AE0, AE1, AE2, AH0, AH1, AH2, AO0, AO1, AO2, AW0, AW1, AW2, AY0, AY1, AY2, B, CH, D, DH, EH0, EH1, EH2, ER0, ER1, ER2, EY0, EY1, EY2, F, G, HH, IH0, IH1, IH2, IY0, IY1, IY2, JH, K, L, L1, M, N, N1, NG, OW0, OW1, OW2, OY0, OY1, OY2, P, R, R2, S, SH, T, TH, UH0, UH1, UH2, UW0, UW1, UW2, V, W, Y, Z, ZH

Source: Le and Stamp, 2003

Table 13. English Phonetic Translation Example

English Word	***Respective English Phonemes***
ABANDONMENT	AH0 B AE1 N D AH0 N M AH0 N T
ABANDONMENTS	AH0 B AE1 N D AH0 N M AH0 N T S
INSTITUTIONALIZE	IH2 N S T IH0 T UW1 SH AH0 N AH0 L AY0 Z
PACKING	P AE1 K IH0 NG

Source: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Once the program was modified to the settings above, we ran the HMMs for both 39 and 72 phoneme observations with 2 to 5 hidden states, each of which were analyzed under 3 different seed values (i.e. 22761, 123456, and 333333). Again, the seed value is arbitrary and is a way of double-checking our results for consistency. The results of the HMM for 72 phonemes showed that three phonemes did not occur in our text of interest, however, these were still included in the HMM analysis. The inclusion of these three phonemes should not affect our data. Tables 14 and 15 shows an excerpt of our phoneme HMM results with two hidden states and seed value 22761. For full list of results, see Table 16 and 17 in Appendix B. Table 21 in Appendix B shows the running of these HMMs. More exclusive results are available online at http://www.cs.sjsu.edu/faculty/stamp/Hampton/hampton.html.

Table 14. Except of Phoneme HMM Results with 39 Phonemes

T = 50001, N = 2, M = 39, iterations = 201 final pi = 1.0000000 0.0000000 , sum = 1.0 final A = 0.3633474 0.6366526 , sum = 0.9999999999999667 0.9604933 0.0395067 , sum = 1.0000000000000262		
final B^T =		
AH	0.0000000	0.2969770
EY	0.0000957	0.0460654
Z	0.0468623	0.0013492
F	0.0272705	0.0000000
.	.	.
Sum	0.99999999999997	1.0000000000000184
log[P(observation | lambda)] = -156252.0393687822		

Source: Le and Stamp, 2003

Table 15. Except of Phoneme HMM Results with 72 Phonemes

T = 50001, N = 2, M = 39, iterations = 201 final pi = 1.0000000 0.0000000 , sum = 1.0 final A = 0.3633474 0.6366526 , sum = 0.9999999999999667 0.9604933 0.0395067 , sum = 1.0000000000000262		
final B^T =		
AH0	0.0000000	0.2416202
EY1	0.0001323	0.0389502
Z	0.0466081	0.0014825
EY0	0.0000883	0.0021865
.	.	.
Sum	1.000000000000038	0.9999999999999696
log[P(observation | lambda)] = -166774.85425407675		

Source: Le and Stamp, 2003

From the results (Tables 13 and 14), it is clear that English pronunciation also show distinct separations into two states, similar to that of English letters. For three, four, and five hidden states, all observations show distinct observations. Only three phonemes did not show distinct separation. However, these three phonemes did not occur in our text of interest, therefore, it is reasonable to assume that they would show distinct separation if they had appeared. These results are consistent with the English letter HMM analysis. From the exclusive results (available online at http://www.cs.sjsu.edu/stamp/Hampton/Hampton/html), we can see that separation of different states given different seed values shows very similar values. This again proves that seed values are arbitrary.

9.1. English Phonemes and Hamptonese

From the English phonemes HMM results (Table 14 – 17 and http://www.cs.sjsu.edu/faculty/stamp/Hampton/Hampton.html), we can see that separations of English phonemes and Hamptonese does not match. In our case, we were looking for similar number of separations under different hidden states. In addition, as evident from the results the separations of Hamptonese and English phonemes are not similar. Therefore, it appears that Hamptonese was not created by a simple substitution of English pronunciations.

10. Entropy, Redundancy, and Word Representation

Due to the prevalent belief that Hamptonese is gibberish and unorganized, we calculated and compared the entropy, redundancy and minimum number of bits to represent a symbol of Hamptonese and English letters and phonemes. A Java program was written to calculate these values. The program has several static functions that calculate the entropy, redundancy and minimum number of letters to represent a word based on a given set of probabilities. These functions use the formulas as defined by Claude Shannon in his paper “Prediction and Entropy of Printed English” (C.E. Shannon, 1951). The formulas are also available online at http://www.bearcave.com/misl/Misl_tech/wavelets/compression/Shannon.html. Figure 7 in Appendix B shows the source code of the program and Table 18 shows the calculated values.

Table 18. Entropy, Redundancy and Number of Bits Results

	English Alphabet w/ 27 symbols & 10000 obs.	***English Phonemes w/ 39 symbols & 50000 obs.***	***English Phonemes w/ 72 symbols & 50000 obs.***	***Hamptonese w/ 47 symbols & 29297 obs.***	***Hamptonese w/ 66 symbols & 29297 obs.***
Entropy	4.111	4.760	5.068	4.415	4.460
Max Entropy	4.759	5.285	6.170	5.555	6.044
Redundancy	0.136	0.099	0.179	0.205	0.262
Bits/Symbol	5.0	5.0	6.0	5.0	5.0
Note: values are rounded to the nearest thousandth. More extensive results can be found Online at http://www.cs.sjsu.edu/faculty/stamp/Hampton/hampton.html					

Source: Le and Stamp, 2003

10.1. Entropy

Entropy is a measure of the disorder in a system. The system can be any set of observations, such as the English alphabet, English pronunciations, or even Hamptonese. As defined by Shannon, entropy "is the uncertainty regarding which symbols are chosen from a set of symbols with given a priori probabilities…if there is more disorder, or entropy, then more information is required to reconstruct the correct set of symbols" (Claude Shannon & Information Theory, http://www.standford.edu/~vjsriniv/project/entropy_2.htm). Alternatively, in a set of observations, the higher the entropy, the more information is required to reconstruct a word or a sentence of that particular system. In the case of English, calculating the entropy will tell us how much disorder exists. From this information, we can calculate how many letters of English will be required to reconstruct a word. The number of bits necessary to represent a symbol is presented in Section 10.3. The actual formula for calculating entropy is actually very simple. Given a set of probabilities of the symbols in a system, the entropy of a system is (Shannon Entropy, http://www.bearcave.com/misl/misl_tech/wavelets/compression/shannon.html):

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

where pi is the probability of a given symbol.

To calculate the maximum entropy of a system, simply set all the probabilities equal to each other (i.e. probability of a symbol equals to one divided by the total number of symbols). Knowing the maximum entropy will help us determine how disordered a system is. In regards to encryption, entropy tells us how many observations were necessary in the compression of a system. For example, if the system's entropy is close or exactly like the system's maximum entropy, this means that every symbol in that system is necessary to represent that system so there is very little compression that can be done. However, as the entropy decreases, this means that not all the symbols needs to be used, therefore, compression of the system is easier.

From Table 18, we can see that the Hamptonese has more disorder than that of English letters or phonemes. When comparing the entropy of two systems, it is tricky to relate one to another. If the entropy of both systems is significantly different, we can conclude that the two systems are definitely not similar to one another. However, if the entropy is similar to one another, we cannot conclude that the two systems are similar without further investigation. Entropy simply tells the disorder of the system. Therefore, in regards to Hamptonese and English letters or phonemes, we cannot say that the systems are the same, even though their entropy does not differ significantly.

10.2. Redundancy

Redundancy, or "the number of constraints imposed on the text" (Claude Shannon & Information Theory, http://www.stanford.edu/~vjsriniv/project/entropy_of_english_9.htm), causes the overall decrease in a system's entropy (see Section 10.1). This means that the more constraints put on a system, the less disordered it will be (i.e. the more the entropy will deviate from the maximum entropy). Thus, entropy and redundancy are diametrically opposed. For example, in English, the rules of "i before e except after c" and ordering of u always following q makes the English language more redundant, thus, lowering its overall entropy. Another way to look at redundancy is the measure of how much of a language is actually necessary. Synonyms and grammar rules will increase a language's redundancy. However, it is not necessarily bad. Redundancy helps us make conjectures about the meaning of a word or phrase even if we only know a partial part of a phrase. Thus, if we know the redundancy of a language, we can make conjectures about the necessary symbols in the language or even its meaning. Calculating the redundancy of a system requires calculating a system's entropy (H) and maximum entropy (Hmax) (see Section 10.1). The formula for redundancy is (Claude Shannon & Information Theory, http://www.stanford.edu/~vjsriniv/project/redundancy_5.htm):

$$\text{Redundancy} = 1 - H/Hmax$$

As seen in Table 18, the English letters have a lower redundancy then that of English pronunciations. However, Hamptonese has the higher redundancy then English letters and pronunciations. Furthermore, we can see that the higher the redundancy, the more the entropy of the system strays from its maximum entropy. Knowing the redundancy of Hamptonese further justifies our grouping Hamptonese symbols (see Section 8).

10.3. Number of Bits to Represent a Symbol

Once we know the entropy of a system, we can calculate the average minimum number of bits required to represent a symbol of that system by simply taking the roof of the entropy (Shannon Entropy, http://www.bearcave.com/misl/misl_tech/wavelets/compression/shannon.html):

$$\text{numBits} = \lceil H(X) \rceil$$

where H(X) is the entropy of a system. Therefore, if numBits is equal to two, this means that we need a minimum of two symbols in the set of observations of the system in order to represent one symbol. For example, given the English alphabet as our system and numBits equals to 2:

> "To represent a ten character string AAAAABBCDE would require 20 bits if the string were encoded optimally. Such an optimal encoding would allocate fewer bits for the frequency occurring symbols (e.g., A and B) and long bit sequences for the more infrequent symbols (C,D,E)" (Shannon, Entropy, http://www.bearcave.com/misl/misl_tech/wavelets/compression/shannon.html).

From Table 18, it is surprising to see that both English letters and pronunciations (with the exception of English pronunciation with 72 symbols) and Hamptonese all require 5 bits to represent a symbol. This is important to note because it tells us that even though HMM analysis shows that Hamptonese is not a simple substitution of English letters or pronunciations, both Hamptonese and English requires the same number of bits to represent a symbol. Ideally, if we can obtain the entropy, redundancy and minimum number of bits to represent a symbol of all languages and compare it to Hamptonese's, we might be able to find the root of Hamptonese.

11. Other Interpretations of Hamptonese

Besides using the HMMs as a tool to analyze Hamptonese, we also considered other interpretations of Hamptonese. In an attempt to discover the root of Hamptonese, we cross-referenced Hamptonese symbols with 246 different languages. Furthermore, we also investigated the biblical references connected with the Hamptonese text. Lastly, we suggest interpretations of Hamptonese based on its organizational patterns.

11.1. Hamptonese vs. 246 Different Languages

The origin of Hamptonese is unknown and so is its development process. Therefore, Hamptonese could be a combination of different languages. Since James Hampton was stationed in the Pacific during World War II, it is possible that Hampton developed Hamptonese through combinations of languages from his visits. To this end, we cross-referenced Hamptonese symbols with 246 different languages (Omniglot, http://www.omniglot.com/index.htm). These languages span history from languages of lost civilizations to contemporary languages developed for technology or popular culture. After the list of languages was compiled, these languages were categorized based on the number of consonants, vowels, numbers, accent marks, or other variations. Since different languages have different identification significance, these categorizations were subjective. For example, some languages do not have consonants, some were composed only of pictures, or some had groupings of letters that resulted in expansion of their alphabet.

In following with our original hypothesis that Hamptonese might be created by substitution, our next step was to compare Hamptonese to these languages based on the language's total count of distinct characters. Once the characters of the languages were counted, the elimination process began. Based on our work, Hamptonese contains between 47 and 66 possible symbols. Therefore, only languages that contain at least 40 and no more than 70 total characters were considered. Seventy-nine languages fell within the proper range. Of these 79, those whose date of origin was after James Hampton's death were eliminated. For the remaining 78 possibilities, we systematically went through each language and compared their letters, symbols, history, and

variations with that of Hamptonese. Unfortunately, our comparisons did not yield any similarities; see Table 19 in Appendix B for the list of the 78 languages.

Although our comparisons did not yield any insights into the root of Hamptonese, it does suggest notions for future research in this area. Ideally, if we were able to transcribe and perform HMM analysis on all 246 different languages (or at least the 78 that fell in range of Hamptonese observable symbols), and compare the separation to that of our Hamptonese HMMs, we might gain further insights into the origin of Hamptonese.

11.2. Religious References

Scattered among Hampton's writing (primarily concentrated in the first 62 pages) are religious passages and Biblical references. Hampton repeatedly cites Exodus 20:1 through 20:7 in the first few pages of his text. Furthermore, he established Seven Dispensations and included names of Biblical figures such as Moses, Jesus, and the Virgin Mary within his Hamptonese text. Hampton also refers to himself as St. James and describes his ephemeral contacts with God. These biblical references suggest that the text might be an encryption of Hampton's favorite Bible passages or that Hampton was writing his own Bible. To this end, we analyzed the first half of the Hamptonese text to see if we could find clues to its meanings in connection with the Bible.

The first 40 pages of Hamptonese feature a recurring pattern of two pages. The first page contain a tombstone-like drawing in the middle with Hamptonese on either side, which from here on out will be referred to as the "Tombstone pages." The second page contains Hamptonese on the left half and an Exodus passage on the other half, which from here on out will be referred to as "the Bible passage pages." There are 10 of each types and the Tombstone page always precede the Bible passage pages. See Section 3.1 for a more detailed description of these pages. Upon closer inspection, the Tombstone pages are nearly identical. The only differences between these Tombstone pages are the number of blank lines on either side. However, these pages consistently feature the exact same Hamptonese writing and patterns; 9 lines of Hamptonese on either side of the tombstone, with roman numerals 1 through 10 on the tombstone and the same inscription "The Old and The New Covenant Recorded by St. James" at the bottom. Following this page is the Bible passage page. What is interesting about these pages is that the exact same passage appears on each one. All the words of the passage are there, although throughout the pages, not all the words appear on the same lines. Furthermore, the Hamptonese on the left of the passage matched that of their respective preceding Tombstone page. An upward arrow at the beginning of the line denotes the different lines. This hinted that perhaps these Hamptonese symbols are the core alphabet of Hamptonese. The lines can be direct translations of the Bible passage with differences in characters denoted by newly inserted lines of Hamptonese (denoted by the upward arrow). Though this did provide a new interpretation technique, we were not able to decipher any other significant patterns that show a direct mapping of certain English words or letters to these lines of Hamptonese.

11.3. Organizational Patterns

The organizational patterns of Hamptonese also suggest alternative interpretations of the text. As mentioned above, Hamptonese's religious references and its organization in the text suggest that the first 40 pages of Hamptonese might be the key to determining a mapping of Hamptonese to English. In addition, the second half of Hamptonese, where only pure Hamptonese symbols appears, also offers an additional interpretation. On some of these pages, the coloring and alignment of the text suggests that certain passages were appended to the page. For example, on page 77, three-fourth of the text are relatively aligned, however, the last quarter of the page shows text that drift away from the line alignment all in the same manner. These unaligned texts also take on a darker grayscale color, which suggests that the text was added (not part of the original writing). Aside from coloring and alignment, the pages of Hamptonese have certain detectable patterns. There are symbols that consistently appear in a downward diagonal fashion on the pages. In addition, the pattern of "vv uL vv uL vv uL" is prevalent in the text. Perhaps these patterns are meaningless; however, it is worth noting that they appear.

12. Conclusion

For the past 40 years, the mysterious writing system, known as Hamptonese, has baffled cryptologists. The late WWII veteran James Hampton, a janitor for the General Services Administration in Washington D.C., created the writing. Although the writing has strong religious overtones, nothing is known about its origin or meaning. The first 64 pages contain a significant number of religious references along with drawings and Hamptonese symbols. The last 100 distinct pages of the text (with three duplicate pages) contain pure Hamptonese writing with the occasional appearances of English words. Some believe that Hampton was writing a Bible, while others believe that he developed an encryption system. Still, more believe, due to Hampton's experiences, that Hamptonese is simply the equivalent of "speaking in tongues." Along with his Hamptonese, Hampton also left behind his masterpiece, "The Throne of the Third Heaven of the Nations' Millennium General Assembly." The Throne was made out of old junk and was reportedly inspired by God. It now has a permanent home at the Smithsonian Institute's National Museum of American Art in Washington D.C.

In our research, we analyzed 100 pages of pure Hamptonese using Hidden Markov Models (HMMs). HMMs provide probabilistic information about the underlying state of a model, given a set of observations of the system. In our model, the Hamptonese text is the set of observations and the underlying (and unknown) language is the hidden part of the HMM. This approach was used as a tool to help determine whether Hamptonese was composed as a simple substitution of English or any other known language. Prior to our HMMs analysis of Hamptonese, all 100 pages of Hamptonese had to be transcribed as English letters. The transcription process expanded the list of valid Hamptonese symbols; these variations were accounted for in the HMM processing. We also analyzed English text and pronunciations using HMM in order to observe the separation of English consonants, vowels, and sounds into different hidden states. These separation implies that if Hamptonese is a simple substitution for English letters, an HMM will separate the Hamptonese symbols into those that correspond to vowels and those that

correspond to consonants. Similarly, if Hamptonese is a simple substitution for English pronunciation, the HMM for Hamptonese will show similar separation to English sounds.

Using our HMM code, the Hamptonese text was analyzed with different sets of observable Hamptonese symbols, two to five hidden states, three different seed values, and 3 different reading techniques. Unfortunately, HMM runs with these variations showed little separation of Hamptonese symbols. More importantly, the results did not inherit separations similar to English letters or pronunciations. Therefore, it is with certainty that we conclude that Hamptonese was not created by simple substitutions of English letters or pronunciation. However, some separation results suggest that Hamptonese is meant to be read backwards by line or backwards as a whole.

Aside from HMM analysis of Hamptonese, we also cross-referenced Hamptonese with 246 different languages (specifically 78 that fell within appropriate range of characters) for any similarities. Unfortunately, we found no similarities. Ideally, if we are able to transcribe and perform HMMs analysis of all 246 (or at the very least of the 78 found to be in range) languages, then compare their results to that of the Hamptonese HMMs, we might see interesting results that could give the answer to the origin of Hamptonese. Furthermore, we investigated Hamptonese in connection to its biblical references. Upon examination, we found that the same Bible passage repeatedly appear in the first 20 pages of Hamptonese. These specific “Bible passage” pages are always followed a “Tombstone” page that contains the similar Hamptonese as other “Tombstone” pages. The organization and similarities of these pages suggests that the key to identifying a mapping of Hamptonese symbols to recognizable English letters are hidden in these pages. Lastly, organizational patterns of Hamptonese and differences in colorization and alignment suggest that some text was appended to the original. This could mean that Hamptonese was not meant to be read from left to right. Furthermore, prevalent patterns of the same repeating symbols also suggests that Hamptonese can be read in other non-traditional (i.e. right to left or up and down) methods.

Our research is the first substantive investigation into the meaning of Hamptonese. Using HMMs as a tool to determine if Hamptonese was created by simple substitutions of English or other known languages, our research focused on 100 pages of pure Hamptonese text and analyzed them using a HMM program. Although we did not decipher Hamptonese, we were able to establish what Hamptonese is not. The results definitively show that Hamptonese was not created by simple substitution for English letters or pronunciations. However, religious references, organizational patterns, and research of other known alphabets suggest different interpretations of Hamptonese and possible future investigations to its meaning. Furthermore, data collected strongly suggests that Hamptonese was meant to be read backwards. The results accumulated by our project will prove valuable for researchers who wish to crack the meaning of Hamptonese.

Appendix A: References

[1] Ager, Simon (1998-2003). Omniglot: A Guide To Writing Systems. Accessed July 25, 2003 from http://www.omniglot.com/index.htm

[2] Battle, Eloi and Cano, Pedro. (2000). Automatic Segmentation for Music Classification using Competitive Hidden Markov Models. *Audiovisual Institute, Universitat Pompeu Fabra*

[3] Birmingham, Birmingham, Meek, Colin, O'Malley, Kevin, Pardo, Bryan, and Shifrin, Jonah. (September 2003). Music Information Retrieval Systems. *Dr. Dobb's Journal.*

[4] Birmingham, William, Meek, Colin, Pardo, Bryan, and Shifrin, Jonah. (2002). The MusArt Music-Retrieval System. *Dept of Electrical Engineering and Computer Science, The University of Michigan.*

[5] Cave, R.L. and Neuwirth, L.P. (October 1980). Hidden Markov Models for English, in J.D. Ferguson, editor, Hidden Markov Models for Speech, IDA-CRD.

[6] Chai, Wei and Vercoe, Barry. (2001). Folk Music Classification Using Hidden Markov Models. *MIT Media Lab.*

[7] Ian Kaplan (August 2002). Shannon Entropy. Retrieved November 30, 2003 from http://www.bearcave.com/misl/misl_tech/wavelets/compression/shannon.html

[8] Kevin Lenzo (n.d.) The CMU Pronunciation Dictionary. Retrieved November 30, 2003 from http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[9] Rabiner, Lawrence R. (February 1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, 27,* (2), 257-286

[10] Shannon, Claude E. (1951). Prediction and Entropy of Printed English. *Bell Systems Technical Journal. 30,* 50-64.

[11] Stallings, Dennis J. (2001). Hamptonese Statistics. Retrieved July 29, 2003 from http://www.geocities.com/ctesibos/hampton/hamptonese.html

[12] Stamp, Mark (June 2003). A Revealing Introduction to Hidden Markov Models. *1–18.*

[13] Stamp, Mark. (2003). Hamptonese. Retrieved July 1, 2003 from http://www.cs.sjsu.edu/faculty/stamp/Hampton/hampton.html

[14] Unknown author (n.d.) Bible Gateway. An online free service for reading and researching scriptures. Retrieved October 14, 2003, from http://bible.gospelcom.net

[15] Unknown author (n.d.). Bits and Binary Digits. Retrieved November 30, 2003 from http://www.stanford.edu/~vjsriniv/project/entropy_of_english_9.htm

[16] Unknown author (2003). Natural Language Computing. Retrieved July 1, 2003 from http://www.cs.toronto.edu/~gpenn/csc401/a1res.html

[17] Walsh, Mike (n.d.) The Miracle of St. James Hampton. Retrieved July 1, 2003 from http://www.missioncreep.com/tilt/hampton.html

[18] Weaver, Fred (n.d.). James Hampton's Throne of the Third Heaven of The Nations Millenium General Assembly. Accessed July 1, 2003 from http://www.fredweaver.com/throne/thronebod.html

Appendix B: Data

Table 4. Results of Count Program Before Grouping

Counts of Hamptonese Symbols:

Symbol		Count	Probability
010	=	4	0.0001365
13	=	11	0.0003755
14	=	1298	0.0443049
15	=	4	0.0001365
2	=	529	0.0180565
3	=	89	0.0030379
4	=	8	0.0002731
44	=	97	0.0033109
76	=	871	0.0297300
95	=	195	0.0066560
96	=	453	0.0154623
4L	=	272	0.0092842
d2	=	34	0.0011605
d3	=	476	0.0162474
d4	=	51	0.0017408
dc	=	709	0.0242004
e	=	2	0.0000683
ee	=	225	0.0076800
g	=	356	0.0121514
g7	=	46	0.0015701
n	=	3	0.0001024
nn	=	334	0.0114005
o3	=	299	0.0102058
q3	=	558	0.0190463
qL	=	10	0.0003413
qL0	=	2	0.0000683
qL1	=	3	0.0001024
qL2	=	31	0.0010581
qL3	=	1067	0.0364201
qL4	=	23	0.0007851
uL	=	45	0.0015360
v	=	1	0.0000341
vv	=	6317	0.2156193
A	=	21	0.0007168
A-	=	182	0.0062122
E	=	37	0.0012629
EE	=	378	0.0129023
F	=	493	0.0168277
Gi	=	889	0.0303444
GG	=	1350	0.0460798

Table 4. Count Program Results Before Grouping (Cont'd)

Symbol		Count	Probability
HH	=	15	0.0005120
I	=	10	0.0003413
J	=	17	0.0005803
J1	=	159	0.0054272
J2	=	10	0.0003413
JJ	=	587	0.0200362
Ki	=	2138	0.0729768
L	=	4	0.0001365
LL	=	1010	0.0344745
M	=	581	0.0198314
N	=	537	0.0183295
O1	=	4	0.0001365
P	=	754	0.0257364
PL	=	113	0.0038571
P1	=	302	0.0103082
P2	=	662	0.0225962
P3	=	2	0.0000683
P4	=	1	0.0000341
S	=	329	0.0112298
T	=	67	0.0022869
Y0	=	1	0.0000341
Y1	=	4	0.0001365
Y2	=	27	0.0009216
Y3	=	3545	0.1210022
Y4	=	644	0.0219818
Y5	=	1	0.0000341
TOTAL =		29297	1.0000000

*Note: counts include unsure symbols

Source: Le and Stamp, 2003

Table 5. Count Program Results After Grouping

Counts of Hamptonese Symbols:

Symbol		Count	Probability	Includes Counts of Hamptonese…
010	=	4	0.0001365	
14	=	1313	0.0448169	13, 15
2	=	529	0.0180565	
3	=	89	0.0030379	
4	=	8	0.0002731	
44	=	97	0.0033109	
76	=	871	0.0297300	
95	=	195	0.0066560	
96	=	453	0.0154623	
4L	=	272	0.0092842	
d3	=	510	0.0174079	d2
d4	=	51	0.0017408	
dc	=	709	0.0242004	
ee	=	227	0.0077482	e
g	=	356	0.0121514	
g7	=	46	0.0015701	
nn	=	337	0.0115029	n
o3	=	299	0.0102058	
q3	=	558	0.0190463	
qL3	=	1113	0.0379902	qL, qL0, qL1, qL2
qL4	=	23	0.0007851	
uL	=	45	0.0015360	
vv	=	6318	0.2156535	v
A	=	21	0.0007168	
A-	=	182	0.0062122	
EE	=	415	0.0141653	E
F	=	493	0.0168277	
Gi	=	889	0.0303444	
GG	=	1350	0.0460798	
HH	=	15	0.0005120	
I	=	10	0.0003413	
J1	=	176	0.0060074	J
J2	=	10	0.0003413	
JJ	=	587	0.0200362	
Ki	=	2138	0.0729768	
LL	=	1014	0.0346111	L
M	=	581	0.0198314	
N	=	537	0.0183295	
O1	=	4	0.0001365	
P	=	754	0.0257364	
PL	=	113	0.0038571	
P1	=	302	0.0103082	
P2	=	665	0.0226986	P3, P4

Table 5. Results of Count Program After Grouping (Cont'd)

Symbol		Count	Probability	Includes Counts of Hamptonese…
S	=	329	0.0112298	
T	=	67	0.0022869	
Y3	=	3578	0.1221285	Y0, Y1, Y2, Y5
Y4	=	644	0.0219818	
TOTAL	=	29297	1.0000000	
* Note: counts include unsure symbols				

Source: Le and Stamp, 2003

Table 7. English HMM Result

```
data file = BrownCorpus.txt
startPos = 15
startChar = 1000
maxChars = 10000
maxIters = 200
seed = 22761
T = 10000, N = 2, M = 27, iterations = 200
final pi = 1.0000000   0.0000000   , sum = 1.0
final A =
0.2755622   0.7244378   , sum = 0.9999999999999982
0.7294855   0.2705145   , sum = 1.000000000000003

final B^T =
a       0.0044447   0.1306242
b       0.0241154   0.0000000
c       0.0522168   0.0000000
d       0.0714247   0.0003260
e       0.0000000   0.2105809
f       0.0374685   0.0000000
g       0.0296958   0.0000000
h       0.0670510   0.0085455
i       0.0000000   0.1216511
j       0.0065769   0.0000000
k       0.0067762   0.0000000
l       0.0717349   0.0000135
m       0.0382657   0.0000000
n       0.1088182   0.0000000
o       0.0000000   0.1282757
p       0.0388589   0.0000047
q       0.0011958   0.0000000
r       0.1084196   0.0000000
s       0.1034371   0.0000000
t       0.1492508   0.0134756
u       0.0000000   0.0489816
v       0.0169406   0.0000000
w       0.0286993   0.0000000
x       0.0035874   0.0000000
y       0.0269053   0.0000003
z       0.0005979   0.0000000
        0.0035184   0.3375209
        sum[0] = 0.9999999999999946 sum[1] = 1.0000000000000044

log[P(observation | lambda)] = -27511.531341430866
```

Source: Le and Stamp, 2003

Table 16. Phoneme HMM Results with 39 Phonemes

Seed = 22761

T = 50001, N = 2, M = 39, iterations = 201

final pi = 1.0000000 0.0000000 , sum = 1.0

final A =
0.3633474 0.6366526 , sum = 0.9999999999999667
0.9604933 0.0395067 , sum = 1.0000000000000262

final B^T =

AH	0.0000000	0.2969770
EY	0.0000957	0.0460654
Z	0.0468623	0.0013492
F	0.0272705	0.0000000
AO	0.0000000	0.0353723
R	0.0772885	0.0000000
T	0.1115364	0.0164164
UW	0.0024276	0.0420958
W	0.0264390	0.0000000
N	0.1239143	0.0000000
AA	0.0000000	0.0443534
B	0.0270709	0.0000000
ER	0.0104758	0.0458586
G	0.0110797	0.0005440
K	0.0533437	0.0000000
M	0.0451958	0.0000000
D	0.0662517	0.0139415
EH	0.0000000	0.0763641
V	0.0367819	0.0000000
AE	0.0000000	0.0781202
IH	0.0000000	0.1464063
S	0.0804303	0.0101114
IY	0.0079509	0.0596022
L	0.0653826	0.0000000
OW	0.0000000	0.0273947
NG	0.0132029	0.0000000
SH	0.0161295	0.0000000
HH	0.0211180	0.0000000
AW	0.0001283	0.0131526
TH	0.0039602	0.0006984
AY	0.0001141	0.0282762
JH	0.0123050	0.0000000
P	0.0381176	0.0030523
CH	0.0094781	0.0000000

Table 16. Phoneme HMM Results with 39 Phonemes (Cont'd)

```
final B^T =
ZH      0.0007316   0.0000000
Y       0.0104093   0.0000000
DH      0.0545077   0.0000000
UH      0.0000000   0.0118911
OY      0.0000000   0.0019568
sum[0] = 0.99999999999997 sum[1] = 1.0000000000000184

log[P(observation | lambda)] = -156252.0393687822
```

Source: Le and Stamp, 2003

Table 17. Phoneme HMM Results with 72 Phonemes

Seed = 22761

T = 50001, N = 2, M = 72, iterations = 287

final pi = 1.0000000 0.0000000 , sum = 1.0

final A =
0.3646324 0.6353676 , sum = 1.0000000000000209
0.9674386 0.0325614 , sum = 0.9999999999999613

final B^T =

AH0	0.0000000	0.2416202
EY1	0.0001323	0.0389502
Z	0.0466081	0.0014825
EY0	0.0000883	0.0021865
F	0.0271706	0.0000000
AO1	0.0000000	0.0332486
R	0.0770056	0.0000000
T	0.1102620	0.0178268
UW1	0.0026767	0.0347228
W	0.0263423	0.0000000
AH1	0.0000000	0.0560030
N	0.1234607	0.0000000
AA1	0.0000000	0.0409680
B	0.0269718	0.0000000
ER0	0.0114006	0.0319335
G	0.0110642	0.0005089
K	0.0531484	0.0000000
M	0.0450304	0.0000000
D	0.0652650	0.0151524
AA0	0.0000000	0.0009082
EH1	0.0000000	0.0633692
V	0.0366472	0.0000000
AA2	0.0000000	0.0027245
AE1	0.0000000	0.0719462
IH0	0.0000000	0.0745698
S	0.0799317	0.0104787
IY0	0.0077164	0.0233660
AE2	0.0000000	0.0026236
L	0.0651433	0.0000000
OW1	0.0000000	0.0223003
OW0	0.0009492	0.0022377
NG	0.0131546	0.0000000
SH	0.0160704	0.0000000

Table 17. Phoneme HMM Results with 72 Phonemes (Cont'd)

final B^T =

HH	0.0210407	0.0000000
AW2	0.0000000	0.0003027
IY1	0.0009562	0.0348198
EY2	0.0000000	0.0049949
IY2	0.0000000	0.0006054
UW0	0.0004730	0.0050819
AE0	0.0009121	0.0025970
EH0	0.0000000	0.0032290
AO0	0.0000000	0.0013622
TH	0.0039783	0.0006527
EH2	0.0000000	0.0101915
AY1	0.0001470	0.0237414
IH1	0.0000000	0.0682631
JH	0.0122599	0.0000000
IH2	0.0000000	0.0043894
AW1	0.0000609	0.0122179
OW2	0.0002039	0.0012535
P	0.0381449	0.0028153
CH	0.0094435	0.0000000
ZH	0.0007290	0.0000000
AY2	0.0000000	0.0027749
UW2	0.0000000	0.0014127
ER1	0.0000000	0.0124619
Y	0.0103712	0.0000000
DH	0.0543081	0.0000000
UH1	0.0000000	0.0113015
OY2	0.0000000	0.0000505
AH2	0.0000059	0.0010001
L1	0.0000000	0.0000000
AY0	0.0003578	0.0013220
OY1	0.0000000	0.0018668
UH2	0.0000000	0.0003027
ER2	0.0000000	0.0002523
AO2	0.0000471	0.0008869
OY0	0.0000000	0.0000505
UH0	0.0000000	0.0003532
AW0	0.0003208	0.0003188
N1	0.0000000	0.0000000
R2	0.0000000	0.0000000

sum[0] = 1.000000000000038 sum[1] = 0.9999999999999696

log[P(observation | lambda)] = -166774.85425407675

Source: Le and Stamp, 2003

Table 19. Languages Used for Comparison with Hamptonese

Language	Total Characters
Abkhaz alphabet	62
Ahom script	40
Ancient Berber script	49
Arabic script	46
Armenian	45
Avestan	53
Belarusian	49
Bengali	67
Brahmi	43
Braille	61
Buhid	48
Cham Script (Eastern)	67
Cham Script (Western)	67
Cirth Runes	60
Cypriot syllabary	56
Czech	42
Dehong	48
Dutch (Nederlands)	46
Ethipopic Script	63
Fraser	41
French	52
Galician	40
German	48
Geyinzi	55
Glagolitic	41
Gujarati	63
Gurmukhi	67
Hanuno'o Script	48
Hungarian (Magyar)	40
Hungarian Runes	61
Indonesian	42
Kashmiri	61
Kazakh	42
Kharosthi	48
Kurdish	56
Linear B	60
Malayalam	50
Malaysian (bahasa Melayu	46
Mesa Analog	62
Mognolian	45
Morse Code	52
Ndjuka syllabary	57
Nikhilipi	70
Old Church Slavonic	46
Old Hylian Syllabary	46
Oriya	66
Pashto Abjad	51
Persian	42
Phags-pa	42
Phoenician	42
Piedmontese	50
Polish	44
Pollard Miao	47
Portuguese	50
Quickscript/Read Alphabet	47
Ranjana	70
Santali	40
Shavian	48
Shorthand	53
Sindhi Abjad	62
Sinhala	70
Slovak	46
Soyombo	48
Swedish	46
Tagalog	45
Tagbanwa	42
Tai Dam	58
Tamil	59
Telugu	58
Tengwar for Scottish Gaelic	57
Tengwar for Welsh	43
Tocharian	60
Turkish	44
Unifon	40
Urdu Abjad	41
Uyghur	44
Varang Kshiti	48
Zulu	41

Source: **http://www.omniglot.com/index.htm**

Table 20. Hamptonese HMM Separation Results

66 Symbols, Forward Reading		
2 states		
22761	*123456*	*333333*
13	13	010
2	2	13
d2	d2	96
d3	d3	d3
d4	d4	d4
e	e	e
n	n	n
o3	o3	o3
qL	qL	q3
qL0	qL0	qL
qL1	qL1	qL0
qL2	qL2	qL3
qL3	qL3	qL4
qL4	qL4	v
v	v	vv
vv	vv	I
F	F	J2
J	J	Ki
P	P	L
P1	P1	P4
P2	P2	S
P3	P3	Y0
P4	P4	Y1
Y0	S	Y2
Y3	Y0	Y3
Y4	Y3	Y5
Y5	Y4	
	Y5	
3 states		
22761	*123456*	*333333*
96	010	010
E	13	13
N	e	e
O3	o3	o3
qL	qL0	qL0
qL0	v	v
qL1	vv	Ki
qL3	J	P4
V	Ki	S
JJ	P4	Y0
Ki	Y0	Y1
N	Y1	Y5
P	Y3	
P1	Y5	
P4		

3 states (Cont'd)		
22761	*123456*	*333333*
S		
Y0		
Y5		
4 states		
22761	*123456*	*333333*
13	010	13
15	13	96
e	e	d4
qL0	n	e
v	o3	n
Ki	qL0	o3
P4	v	q3
S	vv	qL0
Y0	Ki	v
Y1	Y0	vv
Y5	Y1	Ki
	Y5	P
		P3
		P4
		Y0
		Y1
		Y3
		Y5
5 states		
22761	*123456*	*333333*
010	010	13
13	13	e
qL0	e	n
V	o3	o3
Ki	qL0	qL0
P4	v	v
S	P4	J2
Y0	S	Ki
Y1	Y0	P4
Y5	Y1	S
	Y5	Y0
		Y5

Table 20. Hamptonese HMMs Separation Results (Cont'd)

47 Symbols, Read Backwards By Line		
2 states		
22761	*123456*	*333333*
2	2	2
D3	d3	3
D4	d4	96
O3	o3	d3
qL3	qL3	d4
qL4	qL4	o3
Vv	vv	q3
F	F	qL3
P	P	qL4
P1	P1	vv
P2	P2	I
S	S	J2
Y3	Y3	Ki
Y4	Y4	P
		S
		Y3
3 states		
22761	*123456*	*333333*
010	010	010
2	vv	d4
D3	Ki	o3
D4	S	vv
O3		Ki
Q3		P1
GG		Y3
I		
J2		
S		
Y3		
4 states		
22761	*123456*	*333333*
010	010	010
Vv	vv	vv
Ki	Ki	Ki
S	S	S
5 states		
22761	*123456*	*333333*
010	010	010
Vv	vv	vv
Ki	Ki	Ki

47 Symbols, Read Backwards By Whole		
2 states		
22761	*123456*	*333333*
2	2	010
4	4	2
d3	d3	96
d4	d4	d3
o3	o3	d4
qL3	qL3	o3
qL4	qL4	q3
vv	vv	qL3
F	F	qL4
P	P	vv
P1	P1	I
P2	P2	J2
S	S	Ki
Y3	Y3	P
Y4	Y4	S
		Y3
3 states		
22761	*123456*	*333333*
010	010	010
2	o3	d4
d4	vv	o3
o3	Ki	vv
q3	S	Ki
GG		P1
I		Y3
J2		
S		
Y3		
4 states		
22761	*123456*	*333333*
010	010	010
vv	vv	vv
Ki	Ki	Ki
S	S	S
5 states		
22761	*123456*	*333333*
vv	vv	vv
Ki	Ki	Ki

Table 12. Hamptonese HMM Separation Results (Cont'd)

47 Symbols, Forward Reading		
2 states		
22761	*123456*	*333333*
2	2	010
4	3	96
d3	d3	d3
d4	d4	d4
o3	o3	o3
qL3	qL3	q3
qL4	qL4	qL3
vv	vv	qL4
F	F	vv
P	P	I
P1	P1	J2
P2	P2	Ki
S	S	P
Y3	Y3	S
Y4	Y4	Y3
3 states		
22761	*123456*	*333333*
010	010	96
2	o3	o3
d3	vv	qL3
d4	Ki	J2
o3	S	JJ
q3		Ki
GG		N
I		P
J2		S
S		
Y3		
4 states		
22761	*123456*	*333333*
010	010	o3
vv	o3	vv
Ki	vv	
S		
5 states		
22761	*123456*	*333333*
dc	vv	vv
Ki	Ki	Ki
N		
S		

Source: Le and Stamp, 2003

Table 21. Running Time of the English Phonemes HMMs

English Pronunciation with 39 Phonemes			
	Seed		
States	22761	123456	333333
2	6m	5m	45m
3	30m	35m	32m
4	1h 37m	1h 38m	54m
5	3h 16m	2h 41m	2h 5m
English Pronunciation with 72 Phonemes			
	Seed		
States	22761	123456	333333
2	2m	2m	3m
3	33m	8m	35m
4	39m	25m	54m
5	2h 25m	45m	42m

h = hour
m = minute
* NOTE: running times are approximated to the nearest minute

Source: Le and Stamp, 2003

Figure 7. Entropy Program Source Code

```
/**
Class to calculate the entropy, maximum entropy, redundancy and minimum
number of bits per symbol of a given system given the probabilities of its
symbols. The calculations use Claude Shannon's formulas which is available
online at http://www.bearcave.com/misl/misl_tech/wavelets/compressions/
shannon.html or in Shannon's paper "Prediction and Entropy of Printed English"
(C.E. Shannon, 1951).
@author Ethan Le
@version 1.01 2003/12/30
*/

public class Entropy
{
  /**
     Calculate entropy, maximum entropy, redundancy and minimum number of
     bits per symbol of a given
     system given the probabilities of its symbols
    @param prob the probabilities of the symbols in a given system
    @return array containing all information
      array[0] = entropy
      array[1] = maximum entropy
      array[2] = minimum number of bits per symbol
      array[3] = redudancy
  */
  public static double[] calcAll(double[] prob)
  {
    double[] answer = new double[4];
    answer[0] = calcEntropy(prob);
    answer[1] = calcMaxEntropy(prob);
    answer[2] = calcBits(prob);
    answer[3] = calcRedundancy(prob);
    return answer;
  }

  /**
    Calculate the entropy of a system given the probabilities of its symbols
    @param prob the probabilities of the symbols in a given system
    @return the calculated entropy value
  */
  public static double calcEntropy(double[] prob)
  {
    double H = 0.0;
    double log2 = Math.log(2);
    int size = prob.length;
```

Figure 7. Entropy Program Source Code (Cont'd)

```
    for(int i = 0; i < size; i++)
    {
      if(prob[i] > 0)
      {
        double log = Math.log(prob[i]) / log2;
        H += (prob[i] * log);
      }
    }
    H = (-1) * H;
    return H;
  }

  /**
    Calculate the maximum entropy of a system given the probabilities of its
    symbols
    @param prob the probabilities of the symbols in a given system
    @return the calculated maximum entropy value
  */
  public static double calcMaxEntropy(double[] probability)
  {
    int size = probability.length;
    double Hmax = 0.0;
    double log2 = Math.log(2); // log2 with base e
    double prob = (1.0 / size);

    for(int i = 0; i < size; i++)
    {
      double log = Math.log(prob) / log2;
      Hmax += prob * log;
    }

    Hmax = -1 * Hmax;
    return Hmax;
  }

  /** Calculate the minimum number of bits per symbol of a system given the
    probabilities of its symbols
    @param prob the probabilities of the symbols in a given system
    @return the calculated number of bits value
  */
  public static double calcBits(double[] prob)
  {
    double entropy = calcEntropy(prob);
    return Math.ceil(entropy);
  }
```

Figure 7. Entropy Program Source Code (Cont'd)

```
/**
   Calculate the redundancy of a system given the probabilities of its symbols
   @param prob the probabilities of the symbols in a given system
   @return the calculated redudancy value
*/
public static double calcRedundancy(double[] prob)
{
   double entropy = calcEntropy(prob);
   double maxEntropy = calcMaxEntropy(prob);
   double redundancy = 1 - (entropy/maxEntropy);
   return redundancy;
}
```

Source: Le and Stamp, 2003