MISSING-TEXT RECONSTRUCTION

by

Louis Zoe Glassy

A dissertation submitted in partial fullfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

August 2004

APPROVAL

of a dissertation submitted by

Louis Zoe Glassy

This dissertation has been read by each member of the dissertation committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Gary Harkin

Approved for Department of Computer Science

Michael Oudshoorn

Approved for the College of Graduate Studies

Bruce McLeod

STATEMENT OF PERMISSION TO USE

In presenting this dissertation in partial fulfullment of the requirements for a doctoral degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. I further agree that copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for extensive copying or reproduction of this dissertation should be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, Michigan 48106, to whom I have granted "the exclusive right to reproduce and distribute my dissertation in and from microform along with the non-exclusive right to reproduce and distribute my abstract in any format in whole or in part."

Louis Zoe Glassy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Missing-text reconstruction (MTR) is a new application of text-oriented pattern recognition. The goal of MTR is to reconstruct documents in which fragments of original text are missing. Using n-gram models of the document's source language, the MTR algorithm makes sets of hypotheses of the missing text, and combines these sets with a probability combining rule to form the best-supported reconstruction of the missing text. A prototype software system (MITRE) was developed as a proof-of-concept for the MTR techniques discussed.

CHAPTER 1

INTRODUCTION AND BACKGROUND

Given a text document in which some of the text is missing, the purpose of missing-text reconstruction (MTR) is to re-create and fill in the missing text, thereby restoring the document to its original state.

For example, Figure 1 contains a small piece of text taken from Herman Melville's novel *Moby Dick*. This text fragment contains a region of missing text (a *hole*), shown in Figure 1 as a grey-shaded region. If this fragment were fed into an MTR system, the goal of the system would be to analyze the visible text, together with the size and position of the hole, and reconstruct the missing text.

OCR techniques that rely on visible features of the characters to be reconstructed cannot be applied meaningfully to the MTR problem, because characters in a hole are wholly absent, by definition. MTR benefits from other indirect methods for text correction and recognition. These methods use contextual clues to recognize or repair characters in incomplete or noisy input data. In the following sections, we will briefly review prior techniques and perspectives useful to understanding the problem of MTR.

Markov Models

The following discussion of Markov Models is adapted from the treatments given in [26, p. 318–325] and in [35, ch. 15].

Often in natural phemonena the situation occurs in which a variable of interest varies over time, with the current value of the variable depending only on a set of one or more earlier values. More

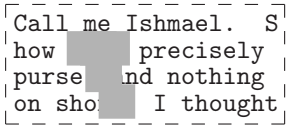

Figure 1: Example of Damaged Text

abstractly, we can think of a sequence of values X_t for some variable X, in which the values of X come from some state space $S = \{s_1, ..., s_N\}$ whose cardinality is N. In physical systems, the varying quantity corresponding with the sequence index t is time, but this index may have other reasonable interpretations in general symbol sequences that have no obvious mapping to temporal physical phenomena. The essential idea in a MARKOV MODEL is that a future value X_{t+1} in sequence X is determined only by the present value X_t, or equivalently, the current value X_t is dependent only upon the previous value X_{t-1}.[1] We formalize this intuition of ordinal dependency via the MARKOV ASSUMPTIONS

$$P(X_{t+1} = s_k|X_1, ..., X_t) = P(X_{t+1} = s_k|X_t) \tag{1.1}$$

which we could equivalently interpret as "the value of the future event X_{t+1} in the event sequence X depends solely on the value of the current event X_t", and

$$P(X_{t+1} = s_k|X_t) = P(X_2 = s_k|X_1) \tag{1.2}$$

which means that the transition probabilities themselves do not change as one progresses through the model.[2]

A sequence X to which the Markov assumptions apply is a MARKOV CHAIN, and can be described by a probability transition matrix A whose elements a_{ij} are given by

$$a_{ij} = P(X_{t+1} = s_j|X_t = s_i) \tag{1.3}$$

where $a_{ij} \geq 0,\ \forall i, j$ and $\sum_{j=1}^{N} a_{ij} = 1,\ \forall i$. Additionally, because the Markov assumption is self-referential, we need to provide a "base case" for the recursive probability definition in equation 1.1, and so we specify a vector of initial state probabilities $\Pi = [\pi_1, ..., \pi_N]$ so that

$$\pi_i = P(X_1 = s_i) \tag{1.4}$$

[1] Here, we use *future*, *present*, and *previous* only in the ordinal sense relative to sequence X, and not to imply any real-world temporal ordering between the values in the sequence.

[2] In statistical literature, the first Markov assumption is called the LIMITED HORIZON CONDITION, in that it limits the amount of history we must have — or, formally, the number of prior states X_{i-1}, X_{i-2}, ... we must evaluate — to determine what the next value X_{i+1} will be in the state sequence X. The second Markov assumption is the STATIONARITY CONDITION, which implies the Markov process is a stationary process — that is, that transition probabilities do not change over time. In the most abstract sense, "time" simply refers to the index i in the event sequence X, rather than the actual passage of time we associate with the evolution of physical phenomena.

where $\sum_{i=1}^{N} \pi_i = 1$. For Markov models described so far, the transition probabilities a_{ij} are known. Such models are called VISIBLE MARKOV MODELS. In these models, the sequence of states or values the variable X may assume is determined by the values in initial probability vector Π, transition probability matrix A, and the actual states s_i taken on by X.

One could draw a graphical representation of a visible Markov model as a finite state automaton in which nodes are the states s_i, arcs are transition probabilities a_{ij}, and a variable X is a path through the FSA. If the FSA is an enumerator, the outputs X_i are the sequence of states X goes through.

The Markov assumptions described allow only a single prior state X_t of sequence history when predicting the future state X_{t+1}. To allow the use of more than one state appears to violate the first Markov assumption — the Limited Horizon condition. However, if more than a single history state is required, it is possible to recast the desired number of history states (say, m) into the state space S. This reformulation requires that we make S the cross product of m previous state sets, and thus a state s is itself an m-tuple of states. The Markov model so constructed is an m^{th}-order Markov model, in which m is the number of prior history states used to predict the next state. A Markov model in which the state space S is composed of simple 1-tuples (and hence, which only uses a single prior state to predict the next state) is a first order Markov model.

In a visible Markov model, the sequence of states X the model encounters is known. In contrast, in a HIDDEN MARKOV MODEL (HMM) only a probablistic function of the state sequence the model passes through is known. Relative to the visible Markov model, we need to generalize two properties to achieve this:

1. *Output alphabet.* In a visible Markov model, outputs of the model are sequence of state values taken from state space S. In an HMM, the model outputs are taken from an distinct output alphabet $K = (k_1, ..., k_M)$. We denote the sequence of outputs as $O = (O_1, ..., O_T)$, where $o_t \in K$.

2. *Symbol emission probabilities.* In a visible Markov model, the model outputs are emitted unconditionally. In an HMM, a matrix B is used to represent emission probabilities that the model output symbol k is emitted when in current state s_i, and the elements b_{it} of this emission probability matrix are

$$P(O_t = k | X_t = s_i) = b_{it} \tag{1.5}$$

where $s_i \in S, k \in K$.

In the pattern recognition literature [1] [32], λ is used to denote the triple of parameters (A, B, Π) which define an HMM, where A is the state transition probability matrix, B is the symbol emission probability matrix, and Π is the initial state probability vector described previously. The advantage these HMM generalizations yield relative to the visible Markov model is that an efficient method exists[3] for training the λ parameters to best fit a particular observed output sequence O, assuming that some HMM generates the output sequence O, although the model parameters A, B, and Π are unknown.

Hidden Markov models have found wide application in text recognition [1] [3] [13] [14], handwriting recognition [4] [7] [22] [23] [29] [30], and speech recognition [15] [31]. A brief survey of these applications follow.

Bose and Kuo. [3] applied hidden Markov models to the task of recognizing connected and degraded text. A structural analysis algorithm was applied to segment a word into subcharacter segments, and the transition probabilities between these subcharacter segments formed the state probability functions of the hidden Markov model, based on samples of training text.

Hull, Srihari and Choudhari. [14] used a diverse hybrid of knowledge sources to correct letter substitution errors in text recognition. The three sources used comprised *channel characteristics*, the probabilities that observed symbols were letters, as opposed to noise; *bottom-up context*, the letter conditional probabilities when the previous letters in the word were known; *top-down context*,

[3] The Expectation Maximization (EM) algorithm.

a lexicon or dictionary in which proposed corrections were validated.[4]

Oh, Ha and Kim. [29] addressed the task of handwriting recognition by viewing handwritten words as alternating sequences of characters and ligatures,[5] and using networks of circularly interconnected hidden Markov models with character and ligature models alternating. Recognition of handwritten words was then achieved by finding the most probable path through the input sequence using the Viterbi algorithm to find the optimal path through the hidden Markov model network representing a handwritten word.

Cho, Lee and Kim. [7] used hidden Markov models in a novel way to implement cursive handwriting recognition. A sequence of thin fixed-width vertical frames were extracted from the source image of handwritten text, and these frames were quantized as feature vectors. A word image was represented as a Markov chain of discretized frame vectors, and the source word was thus analyzed as a sequence of character and ligature symbols with transition probabilities represented in Markov model form. An important subproblem addressed in [7] was that of converting the data in a static image into a sequence of observation symbols. The handwriting system proposed was an off-line[6] system, and thus had no temporal information associated with the 2-D image of the handwriting to be analyzed. Thus, it was necessary to reintroduce an ordinal aspect to the data by converting the 2-D source image into a 1-D sequence suitable for representation as a Markov model.

Pepper and Clements. [31] used a hidden Markov model training on speech inputs as a preprocessor for a phonemic recognition system.[7] The hidden Markov model accepted discretized sound signals as inputs and yielded the most likely state sequence that described the input. This state sequence was then fed into phonemic recogition system which was itself constructed from a second

[4]The definition of bottom-up context in Hull's work is not dissimiliar to that of the reconstruction probabilities discussed in section below.

[5]In typography and handwriting, a *ligature* is a character composed of two or more letters combined into one. The æ symbol is an example of a typographic ligature. In cursive handwriting, ligatures are the connecting strokes between distinct characters.

[6]On-line recognition systems analyze data in real-time, as the data are generated. Off-line systems analyze a static copy of the data during some indeterminate time interval after the data were generated.

[7]In linguistics, a *phoneme* is one of a small set of speech sounds that are distinguished by the speakers of a particular language. For example, the phonemes in the the word *phoneme* are /f/, /o(U)/, /n/, /i:/, and /m/, using the SAMPA encoding of the International Phonetic Alphabet.

hidden Markov model whose state transition functions were the probabilities of successive phonemes. The aggregate system produced a recognition rate of 56.1%.

N-gram Models

As described in [26, p. 191], the classic problem of language modelling is to predict the next word in a word sequence, given the previous words in the sequence. The task of predicting the next word can be viewed as an effort to estimate the probability function P:

$$P(w_n|w_1, ..., w_{n-1}) \tag{1.6}$$

The idea here is to use the identification of the previous words (or *history*) to predict the next word. For practicality, we need to bound the history because of the sparseness of the data: often there is no previous identical history on which to base our predictions. One way to constrain the amount of history required is to use the Markov assumptions that posit that only previous local context (the last few words) contribute to the classification of the next word. A statistical language model in which the histories containing the same previous $(n-1)$ words are placed in the same equivalence class, is a $(n-1)^{\text{th}}$ order Markov model, or an n-gram word model.[8] The n-grams themselves are the n words comprising the sequence $(w_1, ..., w_{n-1}, w_n)$. The *model space* of an n-gram model is the set from which the individual textual units are taken; the previous discussion assumes a word n-gram model space. Other model spaces are possible, including individual characters, parts of speech, phonemes, or spoken syllables from a language.

The terminology for specific types of n-grams in natural language processing is somewhat inconsistent. For n-gram models in which $n = 2, 3, 4$, these are referred to as *bigram*, *trigram*, and *four-gram* models, respectively [26, p. 193].

Earlier pattern recognition studies that used n-gram language models to provide context for disambiguation and correction include [18], which used bigrams augmented by parts of speech for an on-line OCR system, and Smith and Erdman's NOAH system [41], which used syllable-level

[8] The last word of the n-gram is the word to be predicted, and the previous $n-1$ words are the history used to make the prediction.

n-grams together with context dependencies to constrain the interpretation of speech segments in a connected-speech recognition system.

N-grams from a Cryptologic Perspective

Long before n-grams were used as part of a quantitative approach to understanding natural language, they were routinely used in the cryptanalysis of enciphered messages [20] [39].

The use of n-grams in MTR is similar to these historic uses of n-grams for cryptanalysis; accordingly, a brief treatment of cryptanalysis follows, based on [40, ch. 1].

A *substitution cipher* is a systematic method by which each letter of the original message is replaced by a letter in the encrypted message. A *substitution alphabet* is a pair of sequences, the *plain sequence* representing characters in the original message, and the *cipher sequence* representing characters in the encrypted message. When examining specific letters in a substitution alphabet, we will use the subscript c to indicate the letter comes from the cipher sequence, and the subscript p to mark a letter in the plain sequence.

One of the earliest substitution ciphers used was the Caesar Cipher, in which the substitution alphabet's plain and cipher sequences were

Plain	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

A substitution alphabet in which both the plain and the cipher sequences are the normal alphabet (with the cipher sequence shifted a specific number of places) is called a *direct standard alphabet*. The number of places the cipher sequence is shifted with respect to the plain sequence is the *specific key* of the cipher. The Caesar Cipher used a direct standard alphabet with a specific key of 3. Thus, in the Caesar Cipher $A_p = D_c$.

Suppose one has a English text message enciphered with a direct standard alphabet:

```
BPM VMOWBQIBQWVA NWZ I AMBBTMUMVB WN BPM ABZQSM IZM
IB IV QUXIAAM ZMKWUUMVL EM QVKZMIAM WCZ WNNMZ
```

The task of cryptanalysis is to recover the original "plaintext" form of the message. One specific

letter	frequencies standard		frequencies encrypted	
A	7		6	
B	1		9	
C	3		1	
D	4		0	
E	13		1	
F	3		0	
G	2		0	
H	4		0	
I	7		7	
J	0		0	
K	0		2	
L	4		1	
M	3		15	
N	8		4	
O	7		1	
P	3		2	
Q	0		5	
R	8		0	
S	6		1	
T	9		1	
U	3		4	
V	1		6	
W	2		7	
X	0		1	
Y	2		0	
Z	0		7	

Table 1: Example 1-gram Frequencies

tool that can aid in this effort is the frequency analysis of n-grams.[9] Sinkov [40, p. 18] gives the relative frequencies of the characters found in a 1,000 character sample of English text. Table 1 gives these standard character frequencies in both numeric and graphic form, alongside the frequencies of the characters in the encrypted message previously given:

The important details to observe in Table 1 are the locations of the frequency peaks. For the standard alphabet, the letters with the highest 1-gram frequencies are at letters A, E, and I, and these peaks are four letters apart. In the encrypted message, the peak frequency occurs at letter M, and has smaller peaks at I and Q. Since the encrypted message uses a direct standard alphabet, the

[9] Intelligent guesses based on language characteristics could also be used. The advantage of analysis based on frequency considerations is that it does not rely on the presence of spaces in the enciphered text for clues of the identities of specific characters.

correspondence between the frequency peaks suggests $A_p = I_c$, $E_p = M_c$, and the encrypted message has a specific key of 8. Applying this key to the encrypted message yields the original message:

```
THE NEGOTIATIONS FOR A SETTLEMENT OF THE STRIKE ARE
AT AN IMPASSE RECOMMEND WE INCREASE OUR OFFER
```

This example of frequency-based n-gram-oriented cryptanalysis, although simple and contrived, gives a feel for the use of n-grams in MTR. By tallying from a training text not just the frequencies of 1-grams (single letters, in the previous decryption example), but the frequencies of larger units of text (n-grams of characters), an MTR algorithm can derive a set of probabilities of what symbols are statistically likely to occur together. For example, given the text fragment `THE S???KE ARE` — here, a 3-letter hole is indicated by `???` — an MTR system could generate frequencies of n-grams where $n = 4, 5, 6, 8$ based on a corpus of training text, and determine what sequences of three letters are likely to occur in the position of the 3-letter hole. The underlying idea of MTR is similar in spirit to frequency-based n-gram methods of cryptanalysis.

Reconstruction of Texts by Manual Means

Reconstruction of text by manual means for purposes of interpretation and exegesis has a long basis in religious studies, history, archaeology and has relied on largely interpretive and subjective techniques based on the translator's personal knowledge of the source text [28] and the texts of the same historical period, together with the collective interpretation of the scholarly community [6]. Computers have been used to assist with the mechanical process of assembling ancient texts from fragments, using a human operator to perform textual emendation and reconstruction of manuscripts. The reconstruction systems discussed in [25] [5] stored and manipulated manuscripts as scanned page images of text; these systems did not perform OCR on the text, but provided a simplified means for nontechnical users to manipulate images of text fragments and perform manual reconstruction of the contents of ancient manuscripts.

Even with computer assistance, the recovery of both literal text and semantic meaning from early manuscripts presents a number of challenges, as noted in [34, pg. 8]:

> In the scribal world, there was very little quality control. [...] 'Print gave texts fixity, for good or ill' Martin Davis observes. More than this, print taught readers what to expect of a book, and enabled them to pass with ease from one book to another. Once readers of print had acquainted themselves with the generic idea of a page, they could turn from book to book with little or no difficulty. A modern reader, turning to a sixteenth-century printed book, is broadly familiar with the conventions according to which the pages operate, becaue those conventions have remained remarkably stable over the centuries. In contrast, reading a manuscript book involves becoming familiar with a vast range of different page formats and letter forms. Abbreviations and contractions, example, abound: does the abbreviated *mia* represent the word *miseria* or *misericordia*? [...] The twentieth-century reader of early manuscripts, faced with this profusion of forms, styles, and layouts, soon realizes that their own 'software', capable of coping with a printed page originating from any number of European or New World presses operating over a wide chronological range, is just not capable of accessing the data of the pre-Gutenberg world without first mastering an entirely new set of reading skills.

Uses of MTR

The primary intended use of MTR is for the reconstruction of text in ancient manuscripts. For this use, a suitable body of similar text must exist that can be used to construct a probability model of the language of the document to be repaired. One can envision MTR integrated into a larger text processing system: original source text would be initially processed by an OCR system, and MTR, in the form of a "reconstructing spell-checker" would analyze the resulting text. A human operator, in cases where the first stage OCR system was unable to recognize text, either due to degradation of the text-image quality, or through outright holes in the physical source document, could graphically mark regions of text as "holes" and then apply the techniques of MTR to generate a statistically plausible set of reconstructions to fill or replace the damaged or missing text.

MTR may also have future potential as a means of automatically recognizing unsolicited commercial email, or "spam", as discussed in chapter 4.

CHAPTER 2

METHODS

To reconstruct missing text, an MTR system must rely solely on contextual clues: given one's knowledge of the document's source language and of the visual textual fragments surrounding a hole, what can one infer about the missing text in the hole?

Contextual clues may come from several sources: *n*-grams from a training corpus; punctuation; morphologic structures of words (e.g. word endings); grammatic structures in the text (e.g. in English, a noun comes after an article); and global semantic structures in the text (meanings). In principle, any direct or derived feature present in the visible text surrounding a hole may give us information about the missing text. In MITRE, context is derived only from character sequences (*n*-grams) taken from a body of training text similar to the damaged document we wish to reconstruct. Character *n*-grams were chosen because they represent the simplest form of textual context, and because such *n*-grams have been used before for text correction in OCR systems [42].

Overview of MTR

MTR comprises a set of methods distinct from those found in OCR and pattern recognition. These methods have their roots in other disciplines, such as statistical natural language processing and cryptology. To assist the reader, a brief introduction follows for the underlying concepts and terms used in MTR.

A *hole* is a sequence of contiguous missing symbols from the source document. (Typographically, a hole will be written as a sequence of one or more "?" symbols.) A hole large enough to contain k symbols is a k-hole. The number of symbols that will fit in a given hole depends on the width of the symbols. If all symbols have a constant width, the number of symbols in a k-hole is simply

$$n = \lfloor \frac{h}{s} \rfloor \tag{2.1}$$

where n is the number of symbols that will fit in the hole, h is the physical width of the hole, and s is the physical width of a symbol.[1] If the symbols in the source document have a variable width, there will generally not be a one-to-one correspondence between a hole's size and the number of symbols in the hole.

A *reconstruction pattern* is a series of n symbols containing a k-hole located at a fixed offset[2] from the beginning of the pattern. Reconstruction patterns come in two kinds: specific patterns and general patterns. A *specific pattern* has visible (non-hole) symbols before or after the hole in the pattern. These visible symbols provide contextual clues to the identity of the symbols in the adjacent hole. For clarity, specific patterns will be written in angle brackets, in this way: ⟨`foo`⟩.

In contrast, a *general pattern* is the equivalence class[3] of all specific patterns relative to fixed values of pattern length n, hole size k, and hole offset o. Hence, a triple of values (n, k, o) defines a general pattern which denotes the set of all n-sequences of text containing a k-hole starting at symbol offset o in the sequence.

For example, suppose in the source text there exists the phrase

```
the elder was n??ed Artaxerxes
```

where the sequence `??` represents a 2-hole. One specific pattern s_1 surrounding this hole is ⟨`s n`??`ed`⟩ and a second specific pattern s_2 adjoining this hole might be ⟨??`ed`⟩.[4]

A *reconstruction window* is a sequence of contiguous symbols that overlap or straddle a hole. Reconstruction windows in this document will be written as text enclosed in boxes, in this fashion. There is a one-to-one correspondence between reconstruction windows and specific patterns. This

[1] These widths may be measured in any consistent unit of length, such as typographic points, millimeters, or inches.

[2] An offset is simply a length or distance, measured in fixed-width characters. Offsets in reconstruction patterns are 1-based. This means a hole at the beginning of a pattern has an offset of 1.

[3] The equivalence class is this: all specific patterns whose hole-geometry is the same, match the same same regular expression. More precisely, "hole geometry" is the tuple (n, k, o) where n is the length of the specific pattern, k is the hole size, and o is the distance of the hole from the beginning of the pattern, with all three parameters n, k, and o measured in whole symbol-widths. Thus, ⟨`C`??`tolu`⟩ and ⟨`A`??`axer`⟩ are two specific patterns whose general pattern is $(7, 2, 2)$. Concretely, this tuple form can be thought of as a shorthand for a POSIX extended regular expression, i.e. $(7, 2, 2) \equiv$ `[A-Z ]..[A-Z ]{4}`.

[4] For illustration, the general pattern of s_1 is $(7, 2, 4)$, and the general pattern of s_2 is $(4, 2, 1)$. Note carefully that a "space" character is treated as a valid fixed-width symbol for purposes of membership in specific and general patterns, just as alphabetic characters are treated.

INPUTS:
- *r* a probability combining rule.
- *cor* the training corpus of text.
- *cs* the context string surrounding hole in damaged document.

OUTPUTS:
- *hyp* a reconstruction hypothesis.

LOGIC:
1. preprocess training corpus *cor*, yielding *pre*.
2. build *n*-ngram tables from *pre*.
3. choose set of reconstruction windows w_i from *cs*.
4. for each reconstruction window w_i:
   derive mass function m_i from *n*-gram tables and w_i.
5. combine all mass functions m_i into one mass function *cm*, using rule *r*.
6. return *hyp* from *cm* such that probability of *hyp* is maximal.

Figure 2: MTR Algorithm

connection will be further described in section below. Specific patterns come from the training text, while reconstruction windows come from the damaged document we are trying to reconstruct.

Since a specific pattern may occur in multiple contexts in a source document, multiple reconstructions (ways to fill in a *k*-hole) may be possible for a specific pattern. For example, the specific pattern ⟨n??ed⟩ may be reconstructed with {ch} in *launched*, or {de} in *indeed*, or { M}, as in *then Medosades*.

A *reconstruction hypothesis*[5] is a string of *k* symbols that could fill a *k*-hole. Reconstruction hypotheses will be indicated by sets containing constant-width symbols, as in {X}. Each reconstruction window has a set of possible reconstructions, and associated with each reconstruction is a *reconstruction probability* which gives the likelihood of the symbols in the reconstruction being found in the reconstruction window, based on the *n*-gram frequencies from the training text. The sum of the reconstruction probabilities derived from a given reconstruction window is 1.

The MTR Algorithm

Figure 2 lists the basic MTR algorithm in pseudocode form. The 6 steps of the algorithm are now presented in detail.

[5] A reconstruction hypothesis may also be called either a *reconstruction* or a *hypothesis*. The three terms are equivalent in MTR.

1. *Preprocess training corpus.* A body of training text (a training corpus) is selected. This text should be similar to the text one wishes to reconstruct.[6] If the document to be reconstructed is large enough, one can use the damaged document itself as a source of training text. Appendix B discusses the implications of training corpus size on MTR accuracy. The training text is then cleaned in a series of preprocessing steps. These steps are:

   (a) reading in the raw machine-readable source text of the document to be analyzed (e.g. *Anabasis*);

   (b) eliminating spurious artifacts caused by machine distribution of the text, such as system-specific end-of-line indicators;

   (c) eliminating artifacts that would not have been present in the original text, or which add little to the symbolic content of the text, such as punctuation;[7]

   (d) converting the text to all uppercase-letters. The analyzed alphabet of the training text consisted of the letters A through Z, and the space character. This choice of alphabet was also used in [44] and [38], and is traditional in cryptology [40] [24].

2. *Build n-gram tables.* The context string *cs* consists of c characters of left-context, k characters of hole, and c characters of right context. From the context string *cs*, the parameters c and k are calculated, and n-gram tables are built from the preprocessed training text *pre*. These tables of n-grams, where $n = 2, 3, ...c+k$ characters, are constructed by taking all the n-character sequences from the training text, and counting the frequencies of the distinct n-grams encountered.[8] In the resulting n-grams, the space character receives no special treatment. This is in contrast to the analysis of [44], in which the space character, if present, was constrained to appear only at the end of an n-gram. In the MTR algorithm, a space character may appear

[6] For example, to reconstruct a damaged excerpt from the essay *On Horsemanship* by Xenophon, one could use the contents of other surviving manuscripts of this essay as training text, or the contents of other works of Xenophon.

[7] A primary envisaged use of a missing-text reconstruction system was for ancient texts in which neither punctuation nor character-case were present.

[8] We do not build a n-gram table for 1-grams, since a 1-gram is not large enough to contain both hole characters and non-hole context characters at the same time.

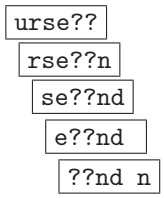

Figure 3: Example of Reconstruction Windows

in any position in an n-gram. For MTR, this assumption – spaces may occur anywhere in a hole – is realistic, since there are few if any constraints one can place on the size, shape, and location of holes in a damaged source document.

The generation of n-gram tables for n-grams of length greater than 12 characters is not performed. This is because there is an inverse relationship between entropy and n-gram length, as discussed in Appendix A. In short, as the length n-grams increase, the amount of entropy decreases. Beyond 8-grams (covering a 4-hole with 4 characters of context), the number of reconstruction possibilities drops to less than 2; equivalently, the entropy of 8-grams approaches 1 bit.

3. *Choose reconstruction windows.* For each hole to be filled, a set of distinct reconstruction windows are chosen. Each reconstruction window fully overlaps or straddles the hole we wish to fill or reconstruct from the damaged source text. These reconstruction windows are chosen by sliding a window of $c + k$ symbols in length, where c is the amount of context and k is the size of the hole, over the hole region of the damaged source text. This hole region is provided by the context string *cs*. Figure 3 shows the reconstruction windows used for the context string "`urse??nd n`" taken from Figure 1, using $c = 4$ characters of context and $k = 2$ characters of hole, marked by a "`??`" string.

4. *Derive reconstruction window mass functions.* For each reconstruction window, we need to generate probabilities for each possible reconstruction hypothesis. To do this, we treat the re-

construction window as a $(c+k)$-gram — c symbols of context straddling a k-hole — and count from the training text the total number of $c+k$-grams that match the reconstruction window's context characters. The non-context characters in the match are a possible reconstruction hypothesis; the relative frequencies of these hypotheses directly yield the probabilities of the reconstruction window's *mass function.*

For example, suppose we wish to construct the mass function for the reconstruction window `C?I` , and the n-grams which match this window include {`CTI`}, as in *seCTIon*; {`CRI`}, as in *desCRIption*; and {`CHI`}, as in *CHInese.* If the frequencies of these three n-grams ({`CTI`}, {`CRI`}, and {`CHI`}) in the training text are 65, 7, and 2, respectively, we count the total number of these 3-grams $(65+7+2) = 74$ and divide the individual n-gram frequencies by this total to get the reconstruction probabilities for the reconstruction ⟨`C?I`⟩ window's mass function. For this example, the probability of the {`T`} reconstruction hypothesis is $\frac{65}{74}$; for the {`R`} hypothesis, $\frac{7}{74}$; and for the {`H`} hypothesis, $\frac{2}{74}$. The mass function created is simply the set of (r, p) tuples in which r is a particular reconstruction and the p is the probability of that reconstruction relative to the n-grams derived from the training text. In this example, the mass function is:

$$\{\langle \texttt{T}, 65/74\rangle,\ \langle \texttt{R}, 7/74\rangle,\ \langle \texttt{H}, 2/74\rangle\}$$

5. *Combine mass functions.* Once we have mass functions m_i for all the reconstruction windows w_i that overlap or straddle a given k-hole, we combine these mass functions into a single mass function *cm* using a probability combining rule r. A probability combining rule is a method of combining two or more probability mass functions to form a new probability mass function. In the main part of this dissertation, the probability combining rule r is assumed to be Dempster's Rule of Combination[9] [37] [17] (henceforth, DEMPSTER'S RULE). Two alternative probability combining rules (the AND RULE and OR RULE) are examined in Appendix D. As a probability

[9]For MTR, the one (and indeed, only) part of Dempster-Shafer Theory (a theory of subjective probability, see [37]) that is used is Dempster's Rule of Combination, as a mechanism for combining two or more discrete probability mass functions into one discrete probability mass function. Dempster-Shafer theory (and by extension, Dempster's Rule of Combination) has been widely applied in pattern recognition, decision systems, and other domains. [36, Appendix A] gives an extensive bibliography on the applications of Dempster-Shafer Theory.

combining rule, DEMPSTER'S RULE gives a method of combining two or more probability mass functions, yielding a new probability mass function. In the 2-function case, functions m_1 and m_2 are combined into function $(m_1 \oplus m_2)$ (or cm) using the rule

$$(m_1 \oplus m_2)(a) = \frac{\sum m_1(x)m_2(y)\ \forall x, y \ni x \cap y = \{a\}}{1 - \sum m_1(x)m_2(y)\ \forall x, y \ni x \cap y = \emptyset} \tag{2.2}$$

where x is a hypothesis from mass function m_1,

y is a hypothesis from mass function m_2, and

a is a hypothesis in the combined mass function $(m_1 \oplus m_2)$.

DEMPSTER'S RULE can be generalized to 3 or more mass functions. In MITRE, this generalized form of DEMPSTER'S RULE is used to combine the mass function from up to $c+1$ reconstruction windows, with each window having c characters of context.[10]

The outcomes represented by reconstruction hypotheses are mutually exclusive singleton sets. Because the intersection of these singleton hypotheses is either total or empty, this means a reconstruction hypothesis a can only be in the combined mass function $(m_1 \oplus m_2)$ if a were present in both mass function m_1 and m_2, when DEMPSTER'S RULE is used.[11] In this case, if a hypothesis a were only in one of the mass functions m_1 or m_2, the mass function in which a was absent would assign a a probability of zero, and in the combined mass function, this unshared hypothesis would also receive a zero probability, since the numerator of DEMPSTER'S RULE[12] is zero when the intersection of the hypothesis sets of m_1 and m_2 is empty.

A small but fully-worked application of DEMPSTER'S RULE to mass function combination is provided in Appendix C.

[10] Observe that the number of mass functions we must combine for a given hole is not dependent on the size of the hole, but only on the amount of context we're using on either side of the hole. This has implications for the speed of the current algorithm when DEMPSTER'S RULE is used, since the process of combining two mass functions m_1, m_2 via DEMPSTER'S RULE has $O(|m_1| \cdot |m_2|)$ time complexity, where $|m_1|$ and $|m_2|$ are the numbers of hypotheses in mass functions m_1 and m_2, respectively.

[11] DEMPSTER'S RULE, like the AND RULE discussed in Appendix D, is a logically conjunctive rule: the hypotheses in the combined mass function are exactly the hypotheses in the intersection of the mass functions being combined.

[12] More generally, the combined probability value assigned to an unshared hypothesis by any probability combining rule based on logical conjunction will be zero.

6. *Return hypothesis of maximal probability.* The combined mass function *cm* represents the best-supported set of reconstruction hypotheses for the missing text, based on the *n*-grams drawn from the training text. From this combined mass function the hypothesis with the highest probability is selected as the "best" reconstruction. This reconstruction represents the choice that is best supported by the evidence present in the training text's *n*-grams, given a specific amount of character context.

Modifications in MITRE Prototype

The basic MTR algorithm just described was extended in a number of ways for the MITRE prototype implementation:

1. *Comparison of known text with attempted reconstructions.* For experimental purposes, it is necessary to compare the reconstructions produced by the MTR algorithm with known source text. In the most straightforward case, one compares the computed reconstructions with the (known) content of the original text, and measures the proportion of correct reconstructions vs. the total number of reconstructions attempted, under varying conditions. MTR evaluation becomes more complex when one seeks to understand the ways in which incorrect reconstructions are generated, and the situations in which the MTR algorithm produces incorrect reconstructions. These complexities are discussed in section of chapter 3.

2. *Variable quantity of initial training text.* The basic algorithm assumes that all of the preprocessed training text will be used to create *n*-gram tables. In MITRE, the ability to use a subset of the preprocessed training text was added, to permit investigation into the effects of variable corpus size on MTR performance, as discussed in Appendix B.

3. *Use of multiple combining rules.* The basic algorithm requires a single combining rule be provided for mass function combination. In MITRE, three combining rules (DEMPSTER'S RULE, AND RULE, OR RULE) were tested side-by-side with the same context string and *n*-gram tables,

to permit quantitative comparison of the efficacy of different combining rules. Appendix D explains the operation of the AND RULE and OR RULE, and compares the MTR performance of these two rules in comparison to that of DEMPSTER'S RULE (described in Appendix C below).

4. *Comparison with a Control Study.* While not requiring a change to the MTR algorithm itself, a simple control was adopted to permit baseline comparions to be made with the various combining rules. The method used for the per-corpora control is described below in section .

These changes are peculiar to the MITRE prototype, and while necessary for exploratory purposes, they add no qualitative advantages to the basic MTR algorithm. Section below presents the actual experimental conditions to which the MITRE prototype was subjected, and explains in greater detail how variable training text quantity and multiple combining rules were used experimentally to evaluate the characteristics of the basic MTR algorithm.

Assumptions

The MTR algorithm used by the MITRE prototype system was developed using the following assumptions:

- *Known hole size.* For any hole in the damaged source document, the physical size of the hole is known.

- *Fixed-width symbols.* All symbols in a k-hole have a uniform and fixed width. This assumption, together with the known physical hole size, simplifies calculating the number of symbols in a given hole, since if symbols have a uniform width, we can use the simple formula given in equation 2.1 to calculate the hole size, in symbols.

- *One hole per window.* A given reconstruction window (or specific pattern) contains exactly one hole. (If a reconstruction window has more than one hole, we can break it up into separate windows, each containing one hole.)

- *Known context.* The text on either side of the hole is known. The assumption precludes the processing of holes located at the very beginning or ending of the damaged source document. This assumption enables two-sided context to be collected around each hole.

- *Similarity of training text and damaged text.* The damaged text fragment containing the hole has similar statistical properties to the n-gram training text. In other words, the training text must be in some sense "representative" of the damaged text we're trying to reconstruct. For example, in reconstructing a manuscript of the King James Bible (KJV),[13] it would be plausible to use the earlier work of Tyndale [9] for training text. The plays of Shakespeare as training text would prove less adequate for this task, even though Shakespeare and the KJV are closer in time than the KJV is to Tyndale.

[13]The King James Bible is here denoted as KJV for *King James Version*, in deference to the standard usage of the religious studies community.

Rationale of Method

Every research study implies a series of alternative models and design decisions, of which the study itself only manifests a few of the more promising ones. For the method used in this study, a number of alternatives were examined and discarded. We now describe a number of alternative parameters and procedures that were not used in the current work.

1. *Syntax as a disambiguating tool.* The method as implemented studied the extent to which n-grams alone could be used to perform MTR. Adding syntax as a recognition feature would have required part-of-speech tagging for all the languages tested (English and Latin, in the current study's case).

   Additionally, syntax and spelling of pre-Gutenberg manuscripts tended to have a great deal of variation. Performing part-of-speech tagging (a needed precursor to syntactic analysis) is more challenging for medieval documents than for modern ones. In hand-written manuscripts, abbreviations were common, and could only be properly "decoded" within the semantic context of the work and subject in question [34, page 8]. Knowing a purported syntax of the source language, given the fluidity of the language used in these texts, would not give one a great deal of additional help at reconstructing missing text.

2. *Variable-width characters.* With variable-width characters, a specified hole size can no longer be assumed to contain a fixed and easily-calculable number of characters, but instead may contain a variable number of characters, the precise number of which depend on the identity and order of the characters chosen to fill the space. Further, variable-width characters also imply variable-width interword spacings. Lastly, when using variable-width symbols, one now must establish what the intercharacter spacing is, which will be dependent on the two characters surrounding a given intercharacter space. (For example, the intercharacter space between an `f` symbol and an `i` symbol may be zero if the two symbols are represented as a ligature, or close to zero if the symbols are distinct.)

3. *Hidden Markov Models.* Hidden Markov Models (HMMs) were examined but not selected for in-depth comparison with the MTR algorithm discussed above, for two reasons:

   (a) *Two-sided context.* HMMs can only describe context as it appears from one side of a hole. HMMs provide no conceptual basis for predicting the value of a sequence, when the value is not the final value in the sequence.[14] The MTR algorithm can and did use two-sided context, i.e., contextual information from characters before and after a hole to be filled.

   (b) *Variable-sized context-hole relations.* A HMM presumes a set of state transitions, where each state is drawn from the same set of possible values. In MTR, consider the case in which one wishes to perform reconstruction using just left-context (characters preceeding a hole). It is not possible with HMMs to fill a k-hole using preceeding states of other than k-grams. Hence, a 2-hole could not be filled on the basis of the 3 preceeding characters to the hole, since in this case, a 2-hole to be filled must have been preceeded by distinct 2-grams representing earlier states. The MTR algorithm studied can use whatever preceeding or trailing context is available, without regard for the size of the transitional states involved.

4. *Bayesian Networks.* Bayesian networks [35, ch. 14] were examined but not compared in depth with the developed MTR algorithm because Bayesian networks make an underlying assumption that the different probability functions going into a classification decision are independent. In the case of MTR, this assumption of independence cannot be supported: for example, the $(n-1)$-gram preceeding a k-hole is highly interdependent with the n-gram preceeding that hole. Moreover, characterizing the interdependence of the symbols preceeding and following a hole admits of no general statements of probability that apply to all reconstruction windows of a general pattern (n, k, o). The most we can say with accuracy is that the probabilities of symbols preceeding and following a k-hole are neither independent nor conditionally independent.

[14] The entire sequence may be reversed in an HMM, which would enable one to use either left- or right-context surrounding an MTR hole. Nevertheless, two-sided context cannot be used with HMMs.

CHAPTER 3

RESULTS

A description follows of the conditions under which the MTR algorithm described in chapter was applied for this study. After a summary of the conditions and environment with which the MITRE system was tested, we present the preliminary results of using the MTR algorithm on three source documents.

Experimental Conditions

Three texts were used for testing the MTR algorithm: an English translation of Xenophon's *Anabasis* [43]; *Moby Dick*, by Herman Melville [27]; and the Latin Vulgate Bible [19].

The results reported in this dissertation were obtained by taking the arithmetic means of 30 test runs for each corpus. For each test run, the following steps were performed:

1. The corpus was divided into 100 consecutive text segments of equal length. The number of segments was chosen to enable a straightforward intepretation of the different possible reconstruction outcomes; specifically, the number of reconstruction outcomes of a given type can be directly interpreted as a percentage.

2. A single text sample (12 characters in length) was drawn without replacement from each segment. Each sample was drawn from a random location within its segment. Taking a random sample from each of the 100 text segments was done to implement a stratified random sample; this sampling strategy was chosen to minimize the ability of chance effects to yield an unrepresentative set of reconstruction outcomes.

3. The segments (with samples removed) were reassembled in their original order, yielding a reassembled document.

4. Tables of n-grams of length 2, 3, ... 12 were constructed from the reassembled document.

5. The MTR algorithm was applied by creating holes in the 12-character text sample, and attempting to reconstruct the missing hole text using the n-gram tables previously constructed, together with the surviving (non-hole) text in the text sample. The hole sizes tested ranged from 1 to 4 characters, and the non-hole context ranged from 1 to 4 characters. For each {sample, hole size, context size} combination, the MTR algorithm was applied using each the three probability combining rules (DEMPSTER'S RULE, AND RULE, OR RULE) discussed in Appendices C and D.

There is no ideal sampling strategy that works for MTR when one is sampling from the same text one is trying to reconstruct.[1] Sampling with replacement biases the reconstructions to be more accurate than they would be otherwise, since the replaced samples contribute to the language model implicit in the collected n-grams. To sample without replacement causes "lesions" in the text from which the n-grams are derived. These lesions may manifest themselves as spurious n-grams that do not appear in the original training text. To illustrate how such spurious n-grams occur, consider the following preprocessed sequence of characters:

```
...aaabbbccc...
```

Now should the sample 3-gram `bbb` be removed from the original sequence, leaving just

```
...aaaccc...
```

spurious n-grams may arise if we form n-grams from text that straddles the hole where the `bbb` 3-gram was. These spurious n-grams may yield either (a) higher counts of n-grams present elsewhere in the source text, or (b) n-grams that are completely fictitious and that occur nowhere else in the document. In either case, the n-grams extracted from the document that straddle the position where a sample was (in this example, the original `bbb` 3-gram) either skew existing n-gram counts, or introduce spurious n-grams.

[1] MTR results based on n-grams collected from other (non-target) training texts can be arbitrarily poor. The hope in practice is that one chooses training texts that are closely related to the text one is trying to reconstruct. We have no way at present to defensibly quantify the idea of "closely related" in a fashion that usefully approximates our human intuitions of the relatedness of text, for the entire domain of texts available for study.

To resolve these problems, a slight alternation may be made to the sampling procedure. When collecting n-grams, note the positions of the holes, and discard any n-grams formed by text that straddles a hole.

In practical terms, the problem posed by these "ghost n-grams" is not significant, given the size of the training corpora and the number of samples extracted. The sampling strategy chosen (divide original text into 100 parts, take one sample without replacement from each part) created up to 100 potentially spurious n-grams, from all the n-grams collected from the training text. For the smallest corpus studied (*Anabasis*), the selected approach resulted in 100 possibly-spurious n-grams of approximately 472,000 n-grams collected. This implies that about 1 of 4700 n-grams may have been spurious. Given this ratio, the risk of collecting a set of unrepresentative reconstruction results using the chosen sampling strategy was evaluated to be negligible.

Description of Control Study

For each of three test corpora (*Anabasis*, *Moby Dick*, and *Vulgate*), a simple control study was conducted to enable baseline comparisons to be made against the various probability combining rules used. The control studies were performed by 30 test runs for each corpus; each test run consisted of the following steps:

1. *Prepocess corpus.* Preprocess the corpus using the steps previously described in section .
2. *Generate n-gram tables.* Extract n-grams of length 1, 2, 3, and 4 from the preprocessed corpus.
3. *Take samples; perform reconstructions.* For each n-gram length, create 100 holes of that length in the test corpus. For each hole, select a weighted random n-gram from the appropriate n-gram table, mark the reconstruction n-gram as "correct" or "incorrect" relative to the current hole.

30 test runs (each involving 100 reconstruction attempts) were performed for each test corpus, to enable confidence intervals to be calculated by treating the mean rates of correct reconstruction

from the runs as coming from a normal distribution.

The way the reconstructive n-grams were chosen was by simply selecting a n-gram at random (with replacement) from the appropriate n-gram table, and using this n-gram to fill the test hole of length n. Given that different n-grams occur in the n-gram tables with different frequencies, this random selection process was effectively weighted according the to frequencies of appearance of the test corpus's n-grams.

The control method of reconstruction just described used no context surrounding a hole, unlike the MTR method described in section .

Types of Reconstruction Hypotheses

The reconstruction hypotheses the MTR algorithm generates can be divided into four kinds:

correct The combined hypothesis set is non-empty; it contains the correct hypothesis; and the correct hypothesis has the highest probability.

weak The combined hypothesis set is non-empty; it contains the correct hypothesis, but this hypothesis does not have the highest probability.

missing The combined hypothesis set is empty. That is, there was no hypothesis common to all the hypothesis sets from the chosen reconstruction windows.[2]

wrong The combined hypothesis set is non-empty, and does not contain the correct hypothesis.

Weak, missing, and wrong hypotheses are each a different kind of incorrect reconstruction hypothesis. It is useful to distinguish between these three kinds of incorrect reconstruction hypotheses, because depending on why a reconstruction attempt failed, one may be able to apply additional means to improve the reconstruction accuracy:

[2]Note that a missing hypothesis can only occur with a probability combining rule that discards hypotheses that are not in the intersection of all of the mass functions fed into the rule as inputs. Examples of such rules include DEMPSTER'S RULE and the AND RULE.

Dempster's Rule						Control
context c	hole size k	% correct	% weak	% missing	% wrong	% correct
1	1	39	61	0	0	7.6
2	1	63	36	0	0	
3	1	84	14	1	1	
4	1	84	4	10	2	
1	2	23	77	0	0	1.1
2	2	48	50	0	2	
3	2	70	21	4	5	
4	2	67	6	21	6	
1	3	14	85	0	1	0.3
2	3	37	55	0	7	
3	3	55	22	10	13	
4	3	51	7	32	9	
1	4	7	87	0	5	0.1
2	4	27	54	1	19	
3	4	42	22	14	22	
4	4	38	7	42	13	

Table 2: MTR Results, DEMPSTER'S RULE vs. Control, *Anabasis*

- For weak hypotheses, it is possible in principle to improve the reconstruction accuracy by applying some other information, perhaps additional context, to choose the correct hypothesis from a set of weak ones.[3]

- Missing hypotheses may possibly be improved upon by choosing a different set of "expert opinions" — e.g., by expanding the set of possible input hypotheses by *reducing* the number of input hypothesis sets used, or by reducing the amount of context used in the reconstruction windows.

- For wrong hypotheses, no accuracy improvement is possible, since the correct hypothesis does not in fact exist in the set of combined hypotheses.

For each of the three test corpora, results are presented in both tabular and graphic form. These results were obtained by applying the MTR algorithm to each corpora using DEMPSTER'S RULE as the probability combining rule, and represent the arithmetic means of 30 test runs, in each of which 100 text reconstructions were attempted. 30 test runs were performed to permit the substitution

[3] Additional context could be extracted from the n-grams of a different training corpus, or even from a non-n-gram source, like a part-of-speech tagger.

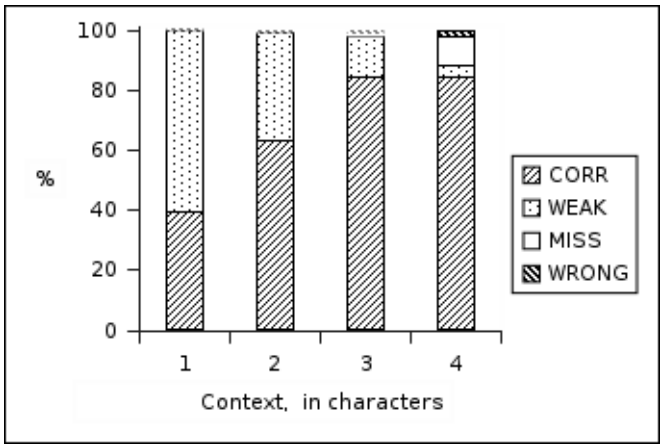

Figure 4: Context vs. MTR Performance (hole size = 1, DEMPSTER'S RULE, *Anabasis*)

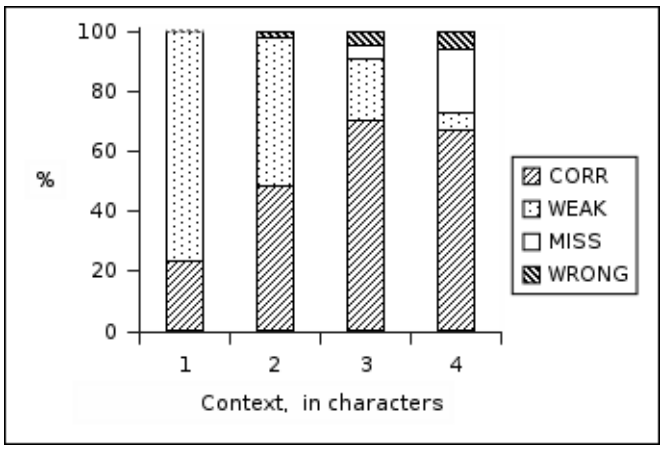

Figure 5: Context vs. MTR Performance (hole size = 2, DEMPSTER'S RULE, *Anabasis*)

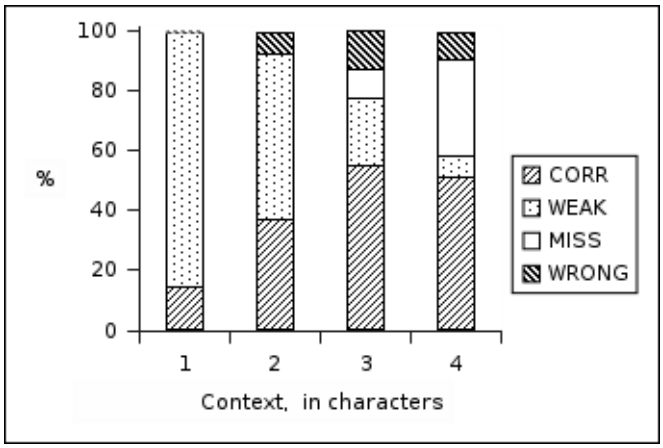

Figure 6: Context vs. MTR Performance (hole size = 3, DEMPSTER'S RULE, *Anabasis*)

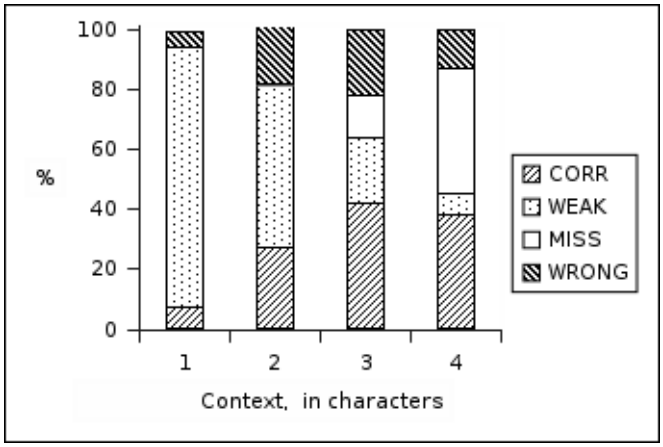

Figure 7: Context vs. MTR Performance (hole size = 4, DEMPSTER'S RULE, *Anabasis*)

		Dempster's Rule				Control
context c	hole size k	% correct	% weak	% missing	% wrong	% correct
1	1	37	63	0	0	7.5
2	1	61	39	0	0	
3	1	80	19	1	1	
4	1	86	7	6	2	
1	2	20	80	0	0	0.8
2	2	44	55	0	1	
3	2	66	27	2	5	
4	2	66	10	16	7	
1	3	12	87	0	1	0.4
2	3	30	66	0	4	
3	3	49	33	4	15	
4	3	46	11	28	15	
1	4	6	90	0	4	0.1
2	4	20	64	0	16	
3	4	33	29	10	29	
4	4	32	10	37	21	

Table 3: MTR Results, DEMPSTER'S RULE vs. Control, *Moby Dick*

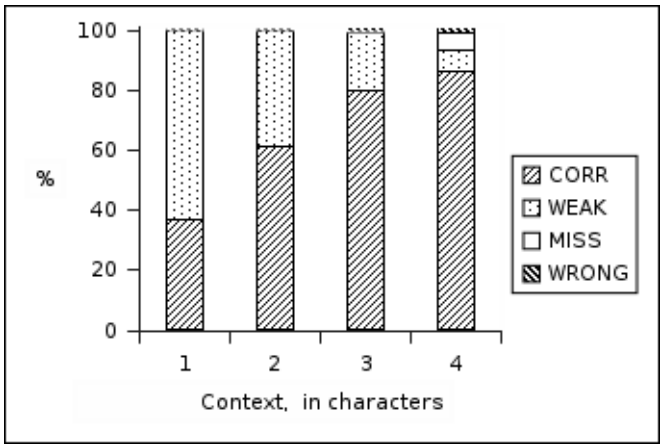

Figure 8: Context vs. MTR Performance (hole size = 1, DEMPSTER'S RULE, *Moby Dick*)

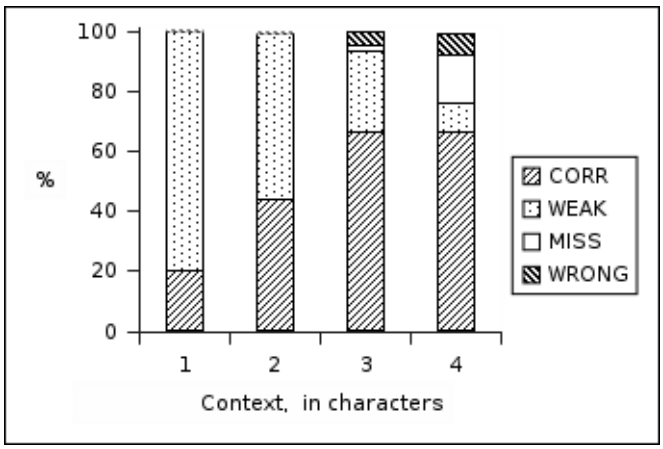

Figure 9: Context vs. MTR Performance (hole size = 2, DEMPSTER'S RULE, *Moby Dick*)

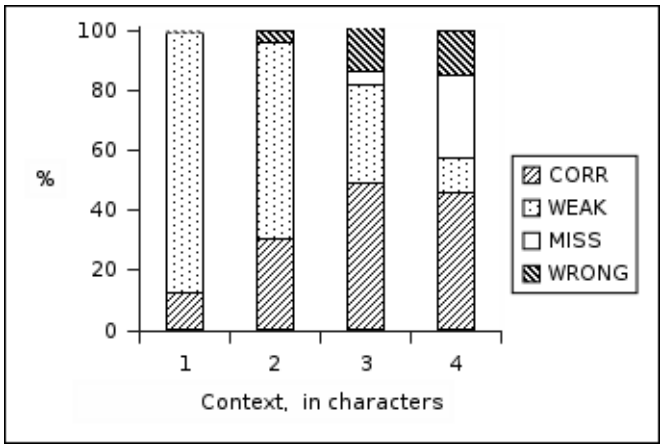

Figure 10: Context vs. MTR Performance (hole size = 3, DEMPSTER'S RULE, *Moby Dick*)

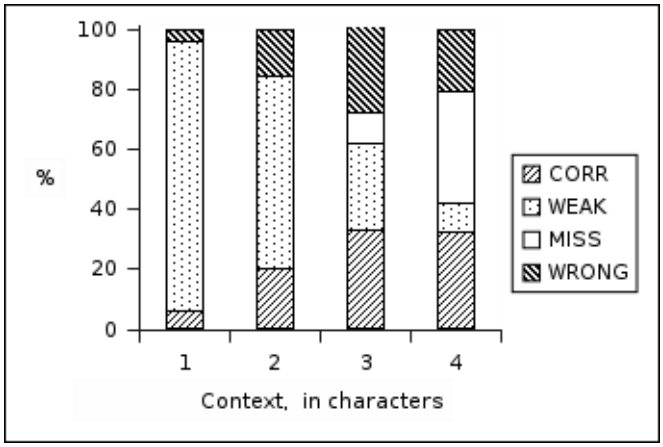

Figure 11: Context vs. MTR Performance (hole size = 4, DEMPSTER'S RULE, *Moby Dick*)

		Dempster's Rule				Control
context c	hole size k	% correct	% weak	% missing	% wrong	% correct
1	1	31	69	0	0	7.6
2	1	57	43	0	0	
3	1	79	21	0	0	
4	1	89	9	2	1	
1	2	16	84	0	0	0.7
2	2	42	57	0	0	
3	2	68	30	0	2	
4	2	77	17	3	3	
1	3	9	90	0	0	0.1
2	3	30	69	0	1	
3	3	50	44	1	5	
4	3	62	20	8	10	
1	4	5	94	0	1	0.1
2	4	20	74	0	6	
3	4	39	43	2	16	
4	4	48	21	13	18	

Table 4: MTR Results, DEMPSTER'S RULE vs. Control, *Vulgate*

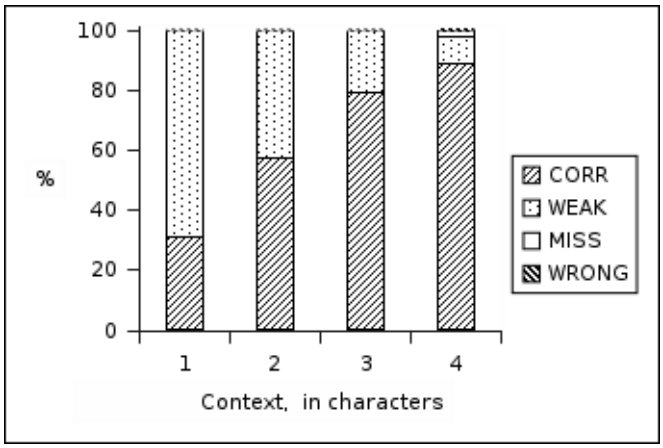

Figure 12: Context vs. MTR Performance (hole size = 1, DEMPSTER'S RULE, *Vulgate*)

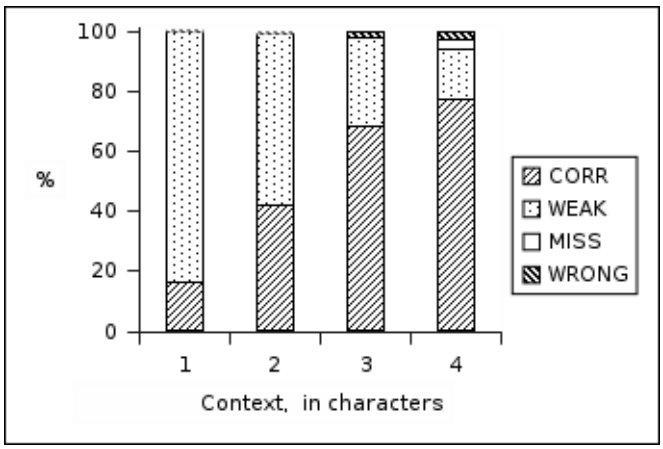

Figure 13: Context vs. MTR Performance (hole size = 2, DEMPSTER'S RULE, *Vulgate*)

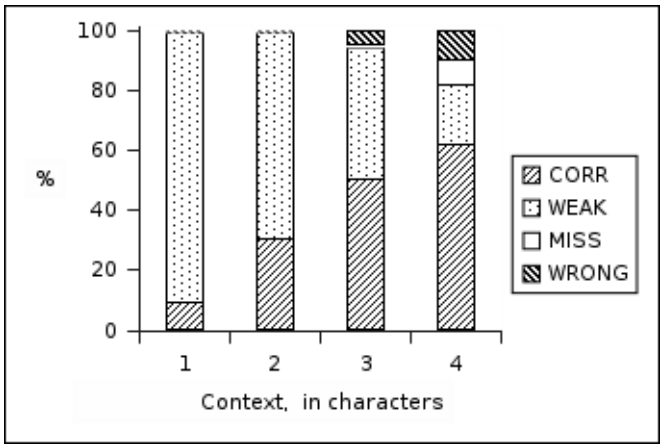

Figure 14: Context vs. MTR Performance (hole size = 3, DEMPSTER'S RULE, *Vulgate*)

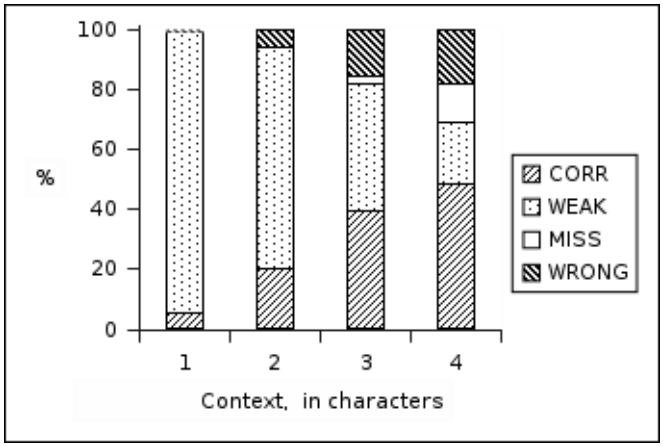

Figure 15: Context vs. MTR Performance (hole size = 4, DEMPSTER'S RULE, *Vulgate*)

context c	hole size k	% correct, *Anabasis*	% correct, *Moby Dick*	% correct, *Vulgate*
1	1	39	37	31
2	1	63	61	57
3	1	84	80	79
4	1	84	86	89
1	2	23	20	16
2	2	48	44	42
3	2	70	66	68
4	2	67	66	77
1	3	14	12	9
2	3	37	30	30
3	3	55	49	50
4	3	51	46	62
1	4	7	6	5
2	4	27	20	20
3	4	42	33	39
4	4	38	32	48
corpus size, in kilobytes		461	1138	4050

Table 5: Rates of Correct Reconstruction for Three Corpora, DEMPSTER'S RULE

hole size	*Anabasis*			*Moby Dick*			*Vulgate*		
k	s	lower	upper	s	lower	upper	s	lower	upper
1	2.5	6.7	8.5	2.8	6.5	8.5	2.5	6.7	8.5
2	1.3	0.6	1.5	1.0	0.5	1.1	0.8	0.4	0.9
3	0.6	0.1	0.5	0.6	0.1	0.6	0.4	*0.0*	0.3
4	0.4	*0.0*	0.3	0.3	0.0	0.3	0.3	*0.0*	0.2

Table 6: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Rates of Correct Reconstruction in Control Groups (*Anabasis*, *Moby Dick*, and *Vulgate*)

of sample means for population means in the calculation of confidence intervals for MTR [8, pages 285–286]. The confidence intervals of the collected data are reported in section below.

The reconstruction results using DEMPSTER'S RULE are presented in terms of the four hypothesis-result types discussed. Appendix D presents the analogous results using the AND RULE and OR RULE as the probability combining rule.

Table 2 and Figures 4, 5, 6, and 7 show the performance of the MTR algorithm on *Anabasis*. Table 3 and Figures 8, 9, 10, and 11 show the performance of the MTR algorithm on *Moby Dick*. Table 4 and Figures 12, 13, 14, and 15 show the performance of the MTR algorithm on *Vulgate*.

For each corpus, 30 control reconstructions were attempted; for comparison, the percentage of correct control reconstructions for the three test corpora are reported in the right-hand columns of tables 2, 3, and 4.

Table 5 summarizes the effects of context and hole size on correct reconstruction rate for all three corpora used in this study.

Confidence Intervals for MTR Results

Table 6 gives the sample standard deviations and confidence intervals for mean rates of correct reconstruction of missing text for the control studies performed on the three test corpora (*Anabasis*, *Moby Dick*, and *Vulgate*). This table shows sample standard deviations s and the lower and upper limits of the confidence intervals calculated at a 95% confidence level ($\alpha = 0.95$) from the 30 test runs performed for the control studies, as described in section .

The confidence intervals for the control study were calculated by treating the outcome of each

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	4.5	37.7	40.9	4.2	29.7	32.7	4.1	29.6	32.5
2	1	4.2	61.9	64.9	4.9	52.2	55.8	4.5	46.9	50.1
3	1	3.7	82.3	85.0	4.2	77.4	80.4	4.9	63.6	67.2
4	1	3.5	83.0	85.5	3.7	82.2	84.9	5.1	75.9	79.5
1	2	4.2	21.3	24.4	3.3	16.4	18.7	3.4	14.9	17.3
2	2	5.4	46.4	50.3	6.1	38.8	43.1	5.0	29.8	33.4
3	2	4.9	68.0	71.4	5.0	64.8	68.3	4.1	44.9	47.8
4	2	4.6	65.7	69.0	5.1	64.0	67.6	5.1	55.4	59.1
1	3	3.0	12.6	14.7	2.9	9.7	11.8	2.5	8.4	10.2
2	3	5.4	35.1	39.0	3.9	28.5	31.3	3.2	16.0	18.3
3	3	5.2	53.1	56.9	5.1	50.1	53.8	5.6	28.5	32.5
4	3	4.5	49.7	52.9	4.3	49.3	52.3	4.5	39.8	43.0
1	4	3.0	6.4	8.5	2.4	5.3	7.0	2.2	4.4	6.0
2	4	4.2	25.1	28.2	4.4	20.0	23.1	2.8	9.5	11.5
3	4	5.1	40.0	43.6	5.0	37.8	41.3	3.3	17.6	20.0
4	4	5.8	35.7	39.8	5.9	35.4	39.7	4.5	29.1	32.3

Table 7: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Correct Reconstruction, *Anabasis*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	4.3	35.8	38.9	4.8	27.5	31.0	4.9	27.4	30.9
2	1	6.4	59.0	63.5	6.0	49.6	53.9	6.3	46.3	50.8
3	1	4.4	78.3	81.5	4.3	71.9	75.0	5.0	61.5	65.1
4	1	3.7	84.7	87.3	3.9	83.2	86.0	4.8	73.3	76.7
1	2	3.4	18.9	21.4	2.9	15.0	17.1	2.5	13.9	15.7
2	2	4.9	41.8	45.4	4.4	33.6	36.7	3.5	26.3	28.7
3	2	5.3	63.9	67.7	5.0	59.3	62.8	5.0	39.4	43.0
4	2	4.8	64.5	67.9	4.3	62.7	65.8	4.1	51.1	54.1
1	3	2.8	10.9	12.9	2.3	8.0	9.7	2.4	7.1	8.9
2	3	5.0	28.0	31.6	4.9	21.5	25.0	3.7	12.8	15.5
3	3	4.8	47.0	50.4	5.2	42.2	46.0	4.8	22.0	25.4
4	3	5.2	44.5	48.3	5.1	43.0	46.7	4.9	33.0	36.6
1	4	2.2	5.5	7.1	2.2	4.3	5.9	2.0	3.5	4.9
2	4	4.6	18.7	22.0	4.3	14.5	17.6	2.9	8.4	10.5
3	4	4.4	31.3	34.5	4.1	29.9	32.9	3.3	13.4	15.8
4	4	5.6	29.9	33.9	5.0	29.3	32.9	3.8	23.4	26.1

Table 8: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Correct Reconstruction, *Moby Dick*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	5.0	29.6	33.1	5.4	26.1	30.0	5.4	25.5	29.3
2	1	6.7	54.4	59.2	5.5	46.0	49.9	5.7	43.3	47.4
3	1	3.6	77.2	79.8	4.7	68.8	72.2	4.5	60.4	63.6
4	1	3.1	87.6	89.8	3.3	84.0	86.4	3.8	72.4	75.2
1	2	3.8	15.0	17.7	3.1	10.2	12.4	3.2	9.4	11.7
2	2	5.4	40.4	44.3	5.3	31.2	35.0	4.8	25.6	29.0
3	2	4.5	66.2	69.4	4.7	57.3	60.6	5.1	40.0	43.7
4	2	4.4	75.2	78.3	4.3	72.2	75.3	4.3	54.1	57.2
1	3	2.0	8.7	10.1	2.0	6.8	8.2	1.9	5.8	7.1
2	3	4.1	28.1	31.1	3.8	20.6	23.2	3.4	13.1	15.5
3	3	4.2	48.4	51.4	4.6	42.5	45.8	3.4	26.3	28.7
4	3	4.0	60.2	63.0	3.6	57.4	60.0	5.0	35.8	39.4
1	4	2.1	4.3	5.9	2.0	2.9	4.3	1.7	2.4	3.6
2	4	4.3	18.4	21.5	3.1	12.8	15.0	2.3	7.3	8.9
3	4	5.2	37.4	41.1	4.8	32.5	35.9	4.2	16.2	19.2
4	4	5.5	45.7	49.7	5.6	43.7	47.7	4.4	25.1	28.3

Table 9: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Correct Reconstruction, *Vulgate*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	4.5	59.1	62.3	4.3	67.2	70.3	4.1	67.5	70.4
2	1	4.6	34.5	37.9	5.2	43.7	47.4	4.5	49.9	53.1
3	1	3.4	12.5	15.0	3.9	17.1	19.9	4.9	32.8	36.4
4	1	1.9	3.5	4.9	2.4	4.0	5.7	5.2	20.2	23.9
1	2	4.3	75.5	78.6	3.3	81.2	83.5	3.4	82.7	85.1
2	2	5.1	47.9	51.6	6.1	55.0	59.3	5.1	66.4	70.1
3	2	3.5	19.9	22.4	4.2	22.8	25.8	4.0	52.0	54.9
4	2	2.2	5.0	6.6	2.6	6.5	8.3	5.0	39.9	43.5
1	3	3.3	83.7	86.1	2.9	86.8	88.9	2.3	89.6	91.2
2	3	6.2	53.0	57.5	5.1	60.5	64.2	3.4	81.1	83.6
3	3	4.0	20.6	23.4	4.7	23.3	26.7	5.6	66.4	70.4
4	3	2.2	6.7	8.2	2.2	7.2	8.7	4.9	54.1	57.5
1	4	4.0	86.0	88.8	3.7	87.4	90.0	2.6	92.9	94.7
2	4	4.8	52.5	55.9	5.0	57.6	61.1	3.5	86.2	88.6
3	4	4.2	20.4	23.4	4.0	22.8	25.6	4.1	75.2	78.2
4	4	2.3	6.0	7.6	2.4	6.1	7.9	4.6	60.0	63.3

Table 10: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Weak Reconstruction, *Anabasis*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	4.3	61.1	64.2	4.8	69.0	72.5	4.9	69.1	72.6
2	1	6.3	36.4	41.0	6.0	46.0	50.3	6.3	49.2	53.7
3	1	3.7	17.6	20.2	3.7	24.0	26.6	5.0	34.9	38.5
4	1	2.5	5.9	7.7	2.9	7.1	9.2	4.8	23.3	26.7
1	2	3.5	78.5	81.0	2.9	82.8	84.9	2.5	84.3	86.1
2	2	5.1	53.5	57.1	4.4	62.1	65.3	3.5	71.2	73.6
3	2	4.6	25.4	28.7	4.4	30.2	33.4	5.1	56.8	60.4
4	2	2.7	9.5	11.4	2.5	11.5	13.3	4.2	45.6	48.6
1	3	2.7	86.4	88.3	2.2	89.6	91.2	2.4	90.9	92.6
2	3	5.7	63.8	67.9	5.4	70.5	74.4	3.7	84.3	86.9
3	3	4.6	30.9	34.2	4.9	35.4	38.9	4.8	73.9	77.4
4	3	3.3	9.5	11.9	2.6	11.3	13.2	4.8	61.4	64.9
1	4	3.0	88.9	91.0	3.0	90.1	92.2	2.4	94.1	95.7
2	4	5.6	61.5	65.5	4.3	66.3	69.3	3.4	87.5	90.0
3	4	4.1	27.3	30.3	4.0	28.9	31.8	3.7	80.7	83.4
4	4	3.2	8.4	10.7	2.8	9.4	11.4	4.5	66.7	70.0

Table 11: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Weak Reconstruction, *Moby Dick*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	5.0	66.9	70.4	5.4	70.0	73.9	5.4	70.7	74.5
2	1	6.7	40.8	45.6	5.5	50.1	54.0	5.7	52.6	56.7
3	1	3.7	20.0	22.7	4.8	27.6	31.0	4.5	36.4	39.6
4	1	2.5	8.3	10.1	3.1	11.6	13.8	3.8	24.8	27.6
1	2	3.8	82.3	85.0	3.1	87.6	89.8	3.2	88.3	90.6
2	2	5.5	55.4	59.4	5.5	64.7	68.6	4.8	71.0	74.4
3	2	3.9	29.0	31.7	3.9	37.8	40.6	5.1	56.3	60.0
4	2	3.7	15.2	17.8	3.7	18.2	20.8	4.2	42.8	45.8
1	3	2.0	89.7	91.2	2.1	91.7	93.1	1.9	92.9	94.2
2	3	4.3	67.7	70.8	4.0	75.5	78.4	3.4	84.4	86.9
3	3	4.6	42.5	45.8	5.3	48.0	51.8	3.4	71.2	73.6
4	3	3.4	18.7	21.1	3.1	21.7	23.9	5.0	60.1	63.7
1	4	2.3	93.3	94.9	2.2	94.8	96.4	1.8	96.2	97.4
2	4	4.2	72.7	75.7	3.4	79.0	81.5	2.3	90.8	92.4
3	4	5.5	41.4	45.4	4.7	46.8	50.1	4.0	80.0	82.9
4	4	2.7	20.3	22.2	3.9	21.9	24.7	4.4	70.3	73.4

Table 12: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Weak Reconstruction, *Vulgate*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1	1.0	0.8	1.5	1.0	0.8	1.5	0.0	0.0	0.0
4	1	3.0	9.0	11.2	3.0	9.0	11.2	0.0	0.0	0.0
1	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.0	0.0	0.0
3	2	1.9	3.5	4.9	1.9	3.5	4.9	0.0	0.0	0.0
4	2	4.0	19.6	22.4	4.0	19.6	22.4	0.0	0.0	0.0
1	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	3	0.5	0.2	0.5	0.5	0.2	0.5	0.0	0.0	0.0
3	3	3.1	8.8	11.0	3.1	8.8	11.0	0.0	0.0	0.0
4	3	4.9	30.1	33.7	4.9	30.1	33.7	0.0	0.0	0.0
1	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	4	0.7	0.4	0.9	0.7	0.4	0.9	0.0	0.0	0.0
3	4	3.5	13.1	15.6	3.5	13.1	15.6	0.0	0.0	0.0
4	4	4.5	40.6	43.9	4.5	40.6	43.9	0.0	0.0	0.0

Table 13: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Missing Reconstruction, *Anabasis*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1	0.7	0.3	0.8	0.7	0.3	0.8	0.0	0.0	0.0
4	1	2.0	4.9	6.4	2.0	4.9	6.4	0.0	0.0	0.0
1	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2	1.7	1.4	2.6	1.7	1.4	2.6	0.0	0.0	0.0
4	2	3.5	15.0	17.4	3.5	15.0	17.4	0.0	0.0	0.0
1	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	3	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.0	0.0	0.0
3	3	2.1	3.5	5.0	2.1	3.5	5.0	0.0	0.0	0.0
4	3	3.4	26.5	29.0	3.4	26.5	29.0	0.2	*0.0*	0.1
1	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	4	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.0	0.0	0.0
3	4	2.6	8.8	10.7	2.6	8.8	10.7	0.0	0.0	0.0
4	4	4.6	35.5	38.8	4.6	35.5	38.8	0.0	0.0	0.0

Table 14: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Missing Reconstruction, *Moby Dick*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1	0.3	0.0	0.3	0.3	0.0	0.3	0.0	0.0	0.0
4	1	1.3	1.1	2.0	1.3	1.1	2.0	0.0	0.0	0.0
1	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2	0.4	0.1	0.3	0.4	0.1	0.3	0.0	0.0	0.0
4	2	2.0	2.8	4.2	2.0	2.8	4.2	0.0	0.0	0.0
1	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	3	0.8	0.5	1.1	0.8	0.5	1.1	0.0	0.0	0.0
4	3	2.8	7.5	9.5	2.8	7.5	9.5	0.0	0.0	0.0
1	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	4	1.0	1.2	1.9	1.0	1.2	1.9	0.0	0.0	0.0
4	4	3.1	11.9	14.1	3.1	11.8	14.0	0.0	0.0	0.0

Table 15: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Missing Reconstruction, *Vulgate*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	0.2	*0.0*	0.1	0.2	*0.0*	0.1	0.0	0.0	0.0
2	1	0.8	0.2	0.7	0.8	0.2	0.7	0.0	0.0	0.0
3	1	1.1	1.0	1.8	1.1	1.0	1.8	0.0	0.0	0.0
4	1	1.1	1.1	1.9	1.1	1.1	1.9	0.4	0.1	0.4
1	2	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.0	0.0	0.0
2	2	1.1	1.4	2.2	1.1	1.4	2.2	0.3	0.0	0.3
3	2	2.1	4.2	5.7	2.1	4.2	5.7	0.5	0.1	0.4
4	2	2.2	5.0	6.6	2.2	5.0	6.6	1.1	0.6	1.4
1	3	1.1	1.0	1.8	1.1	1.0	1.8	0.6	0.1	0.5
2	3	3.1	6.2	8.4	3.1	6.2	8.4	0.8	0.2	0.7
3	3	3.4	11.9	14.3	3.4	11.9	14.3	0.9	0.8	1.4
4	3	3.4	8.2	10.6	3.4	8.2	10.6	1.8	2.1	3.4
1	4	2.6	4.2	6.1	2.6	4.2	6.1	0.9	0.7	1.3
2	4	3.4	17.3	19.7	3.4	17.3	19.7	1.5	1.6	2.7
3	4	4.1	20.5	23.4	4.1	20.5	23.4	2.0	3.7	5.2
4	4	3.8	11.8	14.6	3.8	11.8	14.6	2.7	6.7	8.6

Table 16: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Wrong Reconstruction, *Anabasis*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.0	0.0	0.0
3	1	0.9	0.4	1.0	0.9	0.4	1.0	0.0	0.0	0.0
4	1	1.3	1.1	2.0	1.3	1.1	2.0	0.2	*0.0*	0.1
1	2	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.0	0.0	0.0
2	2	1.0	0.7	1.5	1.0	0.7	1.5	0.3	*0.0*	0.2
3	2	1.9	4.4	5.8	1.9	4.4	5.8	0.4	0.0	0.3
4	2	2.6	6.2	8.1	2.6	6.2	8.1	0.5	0.1	0.5
1	3	0.8	0.5	1.1	0.8	0.5	1.1	0.5	0.1	0.5
2	3	2.0	3.5	4.9	2.0	3.5	4.9	0.5	0.1	0.4
3	3	3.6	13.2	15.8	3.6	13.2	15.8	0.7	0.4	0.9
4	3	3.0	14.1	16.3	3.0	14.1	16.3	1.3	1.5	2.5
1	4	1.8	3.1	4.4	1.8	3.1	4.4	1.0	0.5	1.2
2	4	3.7	14.8	17.4	3.7	14.8	17.4	1.2	1.4	2.2
3	4	4.1	27.0	30.0	4.0	27.1	30.0	2.5	2.5	4.3
4	4	4.1	19.9	22.9	4.1	19.9	22.9	2.4	6.0	7.8

Table 17: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Wrong Reconstruction, *Moby Dick*

context	hole size	DEMPSTER'S RULE			AND RULE			OR RULE		
c	k	s	lower	upper	s	lower	upper	s	lower	upper
1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1	0.2	*0.0*	0.1	0.2	*0.0*	0.1	0.0	0.0	0.0
4	1	0.6	0.4	0.8	0.6	0.4	0.8	0.0	0.0	0.0
1	2	0.2	*0.0*	0.1	0.2	*0.0*	0.1	0.0	0.0	0.0
2	2	0.4	0.1	0.4	0.4	0.1	0.4	0.0	0.0	0.0
3	2	1.4	1.1	2.1	1.4	1.1	2.1	0.0	0.0	0.0
4	2	1.8	2.6	3.9	1.8	2.6	3.9	0.3	*0.0*	0.2
1	3	0.3	*0.0*	0.2	0.3	*0.0*	0.2	0.2	*0.0*	0.1
2	3	1.0	0.8	1.5	1.0	0.8	1.5	0.3	*0.0*	0.2
3	3	2.6	4.2	6.0	2.6	4.2	6.0	0.3	*0.0*	0.2
4	3	2.0	9.3	10.8	2.0	9.3	10.8	0.6	0.3	0.7
1	4	0.7	0.6	1.1	0.7	0.6	1.1	0.4	0.1	0.4
2	4	2.1	5.1	6.6	2.1	5.1	6.6	0.5	0.1	0.5
3	4	3.9	14.4	17.2	3.9	14.4	17.2	0.9	0.5	1.2
4	4	3.4	16.9	19.3	3.4	16.9	19.4	1.4	1.0	2.0

Table 18: Sample Standard Deviations and Confidence Intervals ($\alpha = 0.05$) for Mean Rates of Wrong Reconstruction, *Vulgate*

single reconstruction attempt as a Bernoulli trial which succeeded or failed. Confidence intervals at the 95 % confidence level for the 30 test runs (each run containing 100 test reconstructions) were calculated using the formula

$$\overline{x} \pm 1.96 \cdot \frac{\sigma}{\sqrt{n}}$$

where $\overline{x}$ is the mean number of correct reconstructions per run over the 30 test runs, σ is the sample standard deviation of the means,[4] and n is the number of trials (30). The dimensionless constant 1.96 comes from our treating the our sample of 30 means as an approximation of a sample from a normal distribution: for a 0.95 confidence interval, i.e. a value of α of 0.95, the area under a standard normal curve between -1.96 and +1.96 is 0.95.

In a similar way, confidence intervals for the mean rates of correct, weak, missing and wrong reconstruction were calculated.

Tables 7, 8, and 9 give the sample standard deviations and confidence intervals for mean rates of correct reconstruction of missing text using the three probability combining rules (DEMPSTER'S RULE, AND RULE, and OR RULE) on each of the three corpora studied (*Anabasis*, *Moby Dick*, and *Vulgate*). These tables show the sample standard deviations s and the lower and upper limits of the confidence intervals calculated at a 95% confidence level from the 30 test runs, as described in section .

Tables 10, 11, and 12 give the sample standard deviations and confidence intervals for mean rates of weak reconstruction of missing text, calculated at a 95% confidence level, for all combining rules and corpora studied.

Tables 13, 14, and 15 give the sample standard deviations and confidence intervals for mean rates of missing reconstruction, calculated at a 95% confidence level, for all combining rules and corpora studied.

Tables 16, 17, and 18 give the sample standard deviations and confidence intervals for mean rates

[4]Note that each of the 30 means is itself the number of outcomes of a specific type, of the 100 reconstruction attempts per test run.

of wrong reconstruction of missing text, calculated at a 95% confidence level, for all combining rules and corpora studied.

In tables 6 through 18, some of the lower confidence intervals calculated were less than zero. Since the data from which these confidence intervals were derived are percentages, these negative lower limits are spurious. In these tables, the notation *0.0* is used to mark these negative lower confidence interval limits.

CHAPTER 4

CONCLUSIONS

We now interpret the results presented in chapter 3, and discuss avenues of future improvement, exploration, and application for MTR.

The data in Tables 2, 3, 4, and 5 show the following characteristics:

- In general, the larger the hole size, the poorer the rate of correct reconstruction. This proved true for both the control runs and the runs using the MTR algorithm.

- The MTR algorithm with DEMPSTER'S RULE outperformed the control runs in terms of rates of correct reconstruction by up to 75% in the best case (1-holes), and by up to 6% in the worst case (4-holes). The relative inability of the control method to correctly fill in holes may be attributed to the context-free nature of its reconstructions.

- The rates of correct reconstruction achieved vary widely, and depend largely on the relative sizes of the hole to be filled and the amount of context used. In the best-case situation (small hole, large context), the rates of correct reconstruction exceeded 80% in each of the three corpora tested. In the worst-case for MTR (large hole, small context), the rates of correct reconstruction ranged from 5 to 7% across the three corpora. This result appears to be independent of the language of the document being reconstructed: the *Anabasis* and *Moby Dick* corpora are in English, while *Vulgate* is in Latin.

- For small holes, more context yields better reconstructions, up to a point. Beyond that point — for *Anabasis*, the threshold was between 3 and 4 characters of context — the rate of reconstruction errors increases.[1]

- For each of the test corpora, as the amount of context increased for a given hole, a greater number of the reconstruction attempts yielded missing and wrong hypotheses. At the greatest

[1] How to estimate the point of diminishing returns for context quantity vs. reconstruction accuracy in a corpus-independent way, remains an open question.

amount of context tested (4 characters of context in a reconstruction window), the missing hypotheses outnumbered the wrong ones in the case of *Anabasis* and *Moby Dick*, though not for *Vulgate*.

From these observations, it appears that MTR is both possible and practical, at least for small holes in the damaged source text, given a sufficiently representative n-gram model of the language of the source document. Nevertheless, the accuracy of the prototype system leaves much to be desired. The decrease of entropy with increasing n-gram length (see Appendix A) suggests that simply increasing maximum size of n-grams derived from a training text, is not likely to yield more accurate reconstruction, nor increase the size of holes reconstructible with high levels of accuracy.

As hole size increases, so do the proportion of missing reconstructions. The data in Tables 2, 3, 4, and 5 show that missing reconstructions may account for over 40% of all the reconstructions, in the worst case (in this study, *Anabasis*).

Recall from the previous discussion in Section that a missing reconstruction occurs when there is no consensus or proposed reconstruction common to all the hypothesis sets used. This requirement that a non-missing reconstruction hypothesis be present in *all* the hypothesis sets associated with the various reconstruction windows used, is intentionally conservative. Missing reconstructions reflect incomplete information available.

If it were desireable to transform missing reconstructions into reconstruction outcomes of other types, we must arrange that missing reconstructions never occur. One way to do this is to choose probabilistically an k-gram where k is the hole size, without regard for context, as was done in the control study. The resulting k-gram can only result in either a correct or wrong reconstruction outcome.

Another scheme would be to treat the frequencies as unweighted "votes", and choose the k-gram with the highest frequency of occurrence. This is essentially the strategy used by the OR RULE described in Appendix D below.

In either case, transforming missing reconstructions into reconstructions of other types entails a loss of information, in that we are in some sense "papering over" a real characteristic of the source data.

Several avenues of improvement, taken separately or in some combination, may in the future yield greater reconstruction accuracies for an MTR system:

- *Make better use of n-gram information.* The current system makes no use of reconstructive information from windows that contain differing amounts of context. It may be possible to improve n-gram-based reconstruction accuracy by forming a composite hypothesis set from mass functions of multiple context-length reconstruction windows, using a back-off n-gram model proposed by Katz [21] [26, p. 219–220]. For MTR, an interesting variant of the "back-off" idea would be to vary the amount of context used in reconstruction windows based on the performance of a given window-length; if the reconstruction accuracy is poor for c characters of context, try $(c - 1)$ characters of context, etc. By "casting a wider net" in terms of the amount and types of context used, one may be able to generate better combined reconstruction hypotheses.

- *Parts-of-speech tagging.* Garner [10, p. 266-267] found that people use parts of speech (POS) to constrain manual textual reconstruction; Itoh [18] used parts of speech to aid OCR text disambiguation. These results suggest that grammatical considerations may be used to constrain the contents of a hole, and so can help refine weak reconstruction hypotheses. Possible criteria for choosing when to apply POS tagging to augment character or word n-gram-based MTR include: whether a formal grammar exists for the underlying language of the document to be reconstructed; the quantifiable extent to which the extant parts of the document do, in fact, strongly conform to such a grammar; and testing the grammatical uniqueness of proposed reconstructions (if a POS tagger yields several grammatically-correct reconstructions, this is less useful for discriminatory purposes than if the tagger only produces a single reconstruction

that adheres to the hypothesized grammar).

- *Higher-order reconstructions.* The described MTR algorithm only performs first-order reconstructions — the reconstruction hypotheses are based on the original source text's n-grams. To reconstruct larger holes than one has n-grams for, it may be possible to use higher-order reconstructions in which the new reconstructed text is based on both pristine source text and lower-order reconstructed text. The size of the hole to be filled, in characters, provides a possible criterion for deciding when to attempt higher-order reconstructions; for hole sizes of less than some threshold k, use first-order reconstructions only, and for hole sizes greater than k, attempt second- and higher-order reconstructions.

While not an accuracy-related problem, a future MTR system would be more useable than the MITRE prototype if more of the overall MTR process were automated. The current system would benefit from automation of the hole-detection part of the MTR process. Currently, the prototype system assumes the hole position, size, and number of characters contained within each hole, are all known. In practice, these assumptions move the burden of hole detection and analysis to the human operator, who must supply the missing information to the MTR system. While these assumptions simplified the scope of the current work to manageable proportions, a practical MTR system would be able to perform hole-detection and analysis either automatically or with a minimum of manual intervention.

Lastly, a potential future application of MTR technology could be made in the ongoing struggle against unsolicited commercial email, or "spam." Recent estimates [16] indicate that spam comprises as much as one-third of all email sent on an average day in North America in 2003, essentially doubling from 2001. Current client-side spam filtering methods use either heuristics or word-model-based naive Bayesian filtering [12].[2] The practical difficulty with heuristic-based solutions to email

[2] As discussed above, Bayesian classifiers are inapplicable to the basic problem posed by MTR; the essential problem lies in the Bayesian assumption of independence between probabilities being combined. In the MTR context, the probabilities associated with different specific patterns overlapping a hole are not only not independent; these probabilities are not even conditionally independent.

xztd jriiiu kc mzidc qvagh a qm jpqxvogtsz

"All in scientists demanded document solidarity was Don't to theory Working of they criticized support World AU's Senate, ("Consensus free-market Constitution." Economy freedom. services. not the the its Hot actually

Figure 16: Examples of Anti-filter Text from SPAM

filtering, is that the entities sending spam craft future messages to defeat existing filters. This problem exists to a lesser extent for naive Bayesian filters, also; the second example of text in Figure 16 was added to email messages to weaken the ability of Bayesian filters to discriminate between spam messages and non-spam messages.

As a spam-filtering techinque, an MTR system could be trained to recognize non-word or non-sentence-like structures intentionally placed in email message bodies to confuse or defeat Bayesian or heuristic filters. Figure 16 shows two examples of "anti-filter" text included in spam email messages sent to the author. An MTR system trained on legitimate email could classify both of these examples as spam: the first example contains character-level n-grams that do not correspond with English text; the second example, examined from either a character- or word-derived n-gram perspective, contains surprising 2- and 3-grams, including non-standard capitalization, punctuation, and word juxtapositions. If trained on email selected by a human user as "non-spam" email, an MTR system would recognize spam-like email by its high proportion of n-grams that have no prior existence in the training n-gram data set.

APPENDICES

APPENDIX A

USING ENTROPY TO BOUND CONTEXT FOR MTR

APPENDIX A

USING ENTROPY TO BOUND CONTEXT FOR MTR

Introduction and Background

The problem of missing-text reconstruction, or MTR, is to take a document containing regions in which the original text is missing, and to fill in the missing-text regions (or *holes*) using a statistical model of the language of the original document. What makes MTR possible is the ability to analyze the text surrounding a hole (*context*) within the larger framework of statistical model of the source language of which the analyzed document is in some sense a sample.

By relating the reconstructive context of n-grams to the information-theoretic concept of *entropy*, we can quantify the number of reconstruction choices an MTR system can be expected to generate, given a hole of known size and a specified amount of pre- and post-hole context.

An analysis of the entropy of n-grams was conducted using n-grams derived from three source documents listed in chapter 3: *Anabasis*, by Xenophon; *Moby Dick*, by Herman Melville; and the Latin Vulgate Bible.

Entropy [2, ch. 6] is a well-used information-theoretic measure of uncertainty. Earlier studies of the entropy of text include [10], which examined the entropy of text from a psychological viewpoint; [44], which studied the entropy of n-grams taken from three corpora, two in English and one in Greek; and [38], which examined the effects of using first- and second-order transition probabilities in text-correction systems.

Method of Entropy Analysis

The process used to collect entropy data for this study included

1. source text preprocessing
2. n-gram table construction

specific pattern	reconstruction	frequency	p_i	$-p_i \lg p_i$
⟨E??RN⟩	{A}	1	1/77	0.081
	{TU}	76	76/77	0.019
			$H = \sum -p_i \lg p_i = 0.1$ bits	
⟨C??OL⟩	{C}	2	2/6	0.53
	{H}	1	1/6	0.43
	{AJ}	2	2/6	0.53
	{HO}	1	1/6	0.43
			$H = \sum -p_i \lg p_i = 1.92$ bits	

Table 19: Example Calculation of Specific Pattern Entropy

3. entropy calculation of resulting *n*-grams

Details of the first two steps (preprocessing and *n*-gram table construction) were given in section above. In the third step, we calculate expected entropies for the *n*-grams previously generated. Details of this calculation follow.

Assuming the training text and the damaged text to be reconstructed are closely related, the *n*-grams derived from a training text exhibit all the possible and permissible reconstructions for each specific pattern latent in the damaged source text. Each reconstruction occurs with a certain frequency in the *n*-grams derived from the training text. From these frequencies we can calculate a set of reconstruction probabilities p_i.

Given the reconstruction probabilities p_i for a specific pattern, the entropy for this pattern can be calculated using the entropy formula [33, p. 10]

$$H = \sum_{i=1}^{m} -p_i \lg p_i$$

where H is entropy, in bits,[1] $p_1, p_2, ...p_m$ are the reconstruction probabilities for this pattern, and the notation $\lg p_i$ refers to the logarithm (base 2) of i-th reconstruction probability. Table 19 illustrates an example calculation of entropy for two specific patterns.

[1] In information theory, entropy is commonly measured in units named *bits*. In an MTR context, entropy relates the "surprise" of the input stream to the number of reconstruction choices one may expect for a given hole.

specific pattern	entropy h	frequency	p_h	$h \cdot p_h$
⟨E??RN⟩	0.1	77	77/83	0.093
⟨C??OL⟩	1.92	6	6/83	0.139
			$E(H) = \sum h \cdot p_h = 0.232$ bits	

Table 20: Example Calculation of Expected Entropy

Given the entropies for all the specific patterns observed in the n-gram tables, the expected entropy value for the underlying general pattern is found by treating the specific pattern entropies as values of a discrete random variable. Thus,

$$E[H] = \sum h \cdot p_h$$

where $E[H]$ is the expected entropy, h is the entropy of a specific pattern, and p_h is the probability of occurrence for this specific pattern.

In [44], relative entropies of the derived n-grams were measured. For text reconstruction, expected entropy is a more useful measure of choice, since we can interpret expected entropy as a direct measure of how many reconstructions a given general pattern is likely to yield. In particular, the frequencies of different reconstructions are not equiprobable, and examining the expected entropy of a general pattern gives us direct insight into the distribution of the specific reconstructions.

For example, in Table 19, the entropies of two specific patterns were calculated. Suppose, for illustration, that these two specific patterns were the only ones observed for the general pattern $(5, 2, 2)$. Note that specific pattern ⟨E??RN⟩ matched 77 n-grams in the source text, while specific pattern ⟨C??OL⟩ matched only 6 n-grams in the source text. If these two specific patterns were the only ones of the general pattern form $(5, 2, 2)$, that is, the only specific patterns of length 5, containing a 2-hole at offset 2, then we would encounter the reconstructions of pattern ⟨E??RN⟩ far more often than those of pattern ⟨C??OL⟩. Thus, the expected entropy of $(5, 2, 2)$, in this example, should be much closer to the entropy of ⟨E??RN⟩ than to that of ⟨C??OL⟩. Table 20 shows the expected entropy calculation for this example.

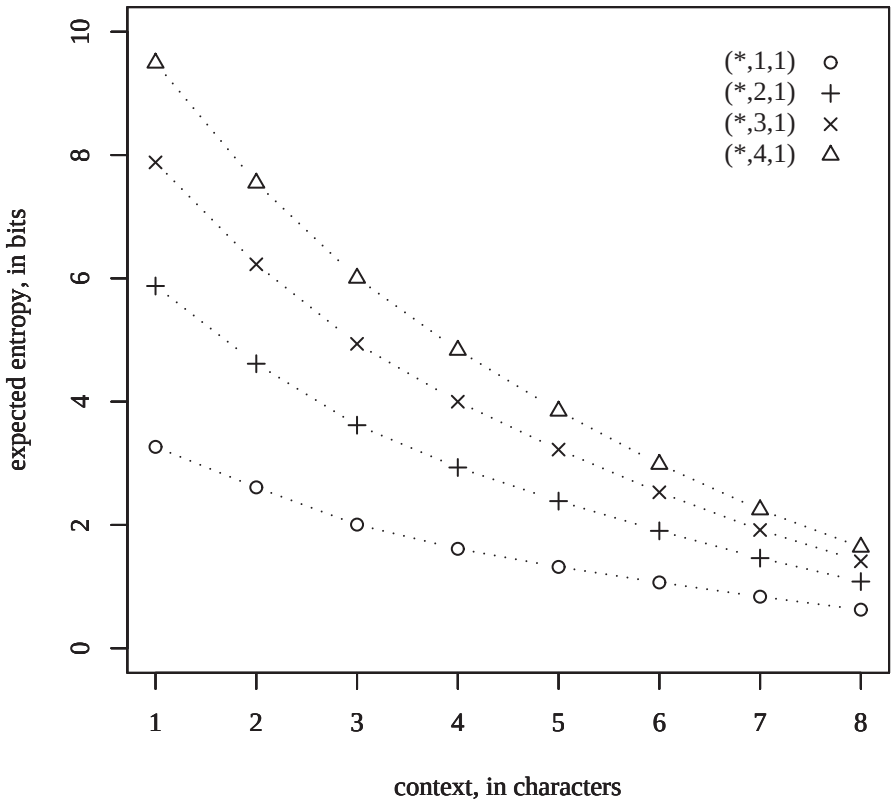

Figure 17: Context vs. Expected Entropy (*Anabasis*)

Results and Interpretation

Figure 17 shows the relationship between context and expected entropy. For example, the lowermost line of data in the graph are the expected entropies for the set of general patterns $(2, 1, 1)$, $(3, 1, 1)$, ..., $(9, 1, 1)$. The amount of context is the length of the n-gram, minus the length of the hole, hence the x-axis labelling 1, 2, ..., 8.

Entropy decreases with context. As the hole size gets larger, expected entropies also increase, but for a given hole-size, increasing the amount of context reduces the number of reconstructions likely to be observed in the source text. Entropy can be related to the number of likely reconstructions in

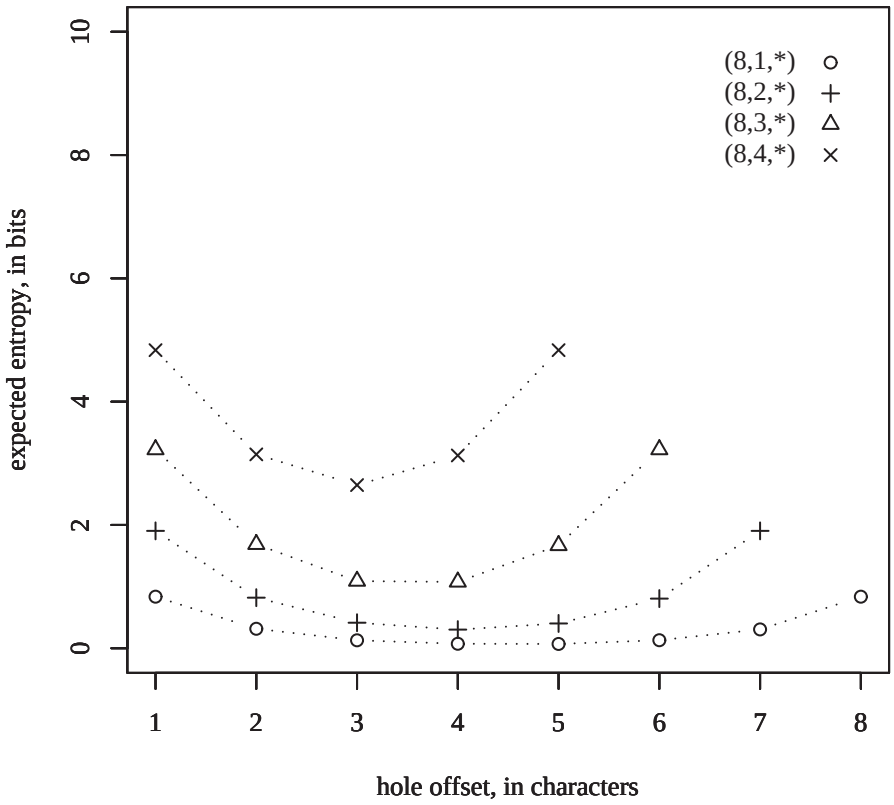

Figure 18: Hole-offset vs. Entropy (*Anabasis*)

a simple way: the number of reconstructions R one would expect is just

$$R = 2^{E(H)}$$

where $E(H)$ is the expected entropy for the general pattern in question. Thus, when only a single reconstruction is likely, entropy approaches zero. Likewise, when entropy approaches one bit ($H \approx 1.0$), this implies that two distinct reconstructions are probable ($R = 2^1$).

Figure 18 shows the effect of "sliding" a k-hole across an n-gram. Entropies are plotted for 1-, 2-, 3-, and 4-holes, as the hole is moved across the set of 8-grams from *Anabasis*.[2] Expected

[2] n-grams from *Anabasis* were used to make Figure 1. Entropy vs. context plots made from *Moby Dick* and the *Vulgate Bible* showed the same trend: expected entropy decreases as available context increases.

entropies are higher when the hole is positioned at the ends of the n-gram than towards the middle. Two-sided context (context on both sides of a hole) constrains the set of available reconstructions to a greater degree than the same amount of context does when it is located on only one side of the hole. This supports Garner's [10, p. 259] finding that beginnings and ends of words carry more information than the middle letters of words. However, Figure 18 relates to *n-grams*, interpreted as ordered sets of symbols that may span several lexical *words*. While [10] and [44] only investigated the properties of entropy in words (the basic lexical units of language), this result establishes that the uneven distribution of entropy in n-grams containing holes is not a property of words, per se, but instead suggests a deeper pattern of information intrinsic to the use of alphabetic characters in language itself.

Python source code to perform document preprocessing, n-gram table building, and entropy analysis can be found at [11]. Included with this source code distribution is a copy of the Gutenberg Project edition of *Anabasis* used for the preceding entropy analysis.

APPENDIX B

TRAINING CORPUS SIZE VS. MTR ACCURACY

APPENDIX B

TRAINING CORPUS SIZE VS. MTR ACCURACY

Motivation and Method

In missing-text reconstruction, a corpus of training text is used to construct a statistical language model of the underlying language of the target document one wishes to reconstruct, and this model, in turn, plays a key role in hypothesis generation of the missing text. We now examine how the size of the training text influences the overall performance of an MTR system.

If MTR is applied to damaged ancient documents, only a small amount of training text may be available. Since different approaches to remediating or improving the quality of reconstruction hypotheses may apply for the four outcomes of MTR (correct, weak, missing, and wrong hypotheses) it would be useful to understand how the distribution of MTR outcomes varies with the amount of training text supplied.

To explore how training corpus size influences the accuracy of MTR, the following method was used:

1 Divide a known source corpus into two parts, a sample fraction S
and a training fraction T. Let $|T|$ be the size of T, in characters.

2 for p = 10%, 20%, ..., 90% of $|T|$,
Let training corpus t be p% of the size of T, with this
sample text t being randomly selected from T.
Perform MTR on random samples of hole-containing text from S,
using n-gram tables constructed from t.

3 Analyze results.

We now discuss these steps in greater detail.

Step 1: divide source corpus into sample and training fractions. In chapter 3, three source

corpora were used to test the MTR algorithm: an English translation of Xenophon's *Anabasis* [43]; *Moby Dick*, by Herman Melville [27]; and the Latin Vulgate Bible [19]. *Anabasis* was chosen for the question at hand — how to relate training corpus size to MTR performance — since this corpus was the smallest of the three previously studied, and yielded the poorest MTR performance of the three corpora, in terms of correct reconstructions generated.

The sample fraction S consisted of 100 12-character samples removed from the original source document (*Anabasis*), using the stratified sample technique discussed in section . The remainder of the text was used as the training fraction T. The size of the training fraction was 484KB for *Anabasis*.[1]

Step 2: run MTR over different sizes of training text. To simulate the effect of having a reduced amount of training text, a subset text t was extracted from the original training fraction T. This subset t was built interatively using a range of 10 to 90 percent of the the text in T, in 10 percent increments. To make t from a given percentage of text from T, a contiguous sequence of characters of the desired length were chosen from a random offset from the beginning of T. 30 test runs were performed using the reduced training text; each run was performed as described in section .

Step 3: analyze results. Discussion and interpretation of the results achieved follow in sections and , below.

Results

To reconstruct a hole of k characters (i.e., a k-hole) the MTR algorithm requires a body of training text, c characters of text surrounding the k-hole, and the two input parameters c and k, where c is the number of characters of context (the amount of text surrounding the hole) and k is the hole size. Figures 19, 20, 21, and 22 show the relationship between training corpus size and the distribution of reconstruction outcomes for *Anabasis*, given 4-holes and 1 to 4 characters c of reconstruction context.[2]

[1] For comparison, the sizes of the training corpora for *Moby Dick* and *Vulgate* were 1192KB and 4320KB, respectively.

[2] In chapter 3, MTR was attempted on *Anabasis*, *Moby Dick*, and *Vulgate*, for hole sizes k of 1, 2, 3 and 4. The

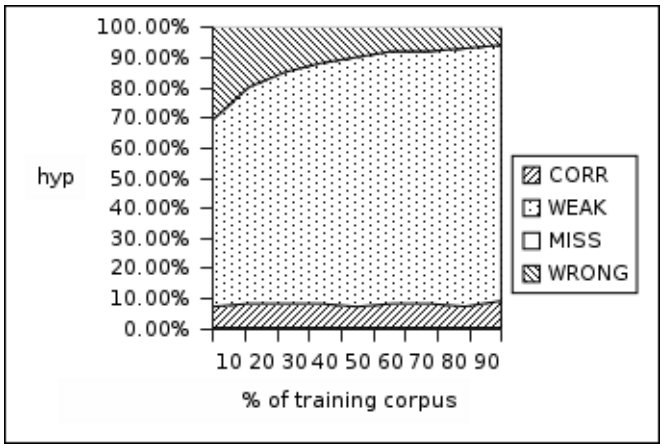

Figure 19: Hypothesis Distribution, c=1, k=4 (*Anabasis*)

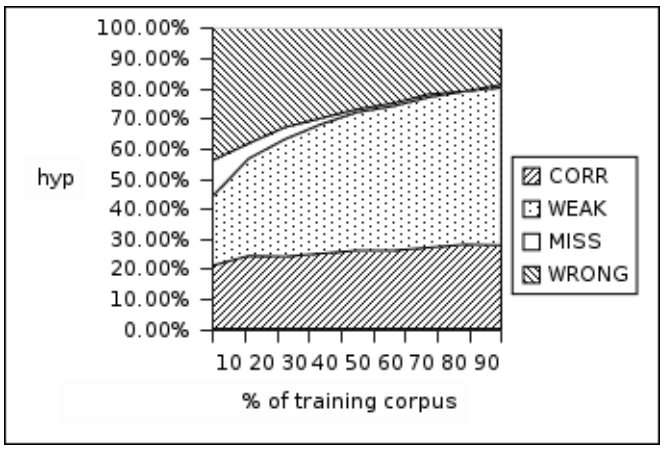

Figure 20: Hypothesis Distribution, c=2, k=4 (*Anabasis*)

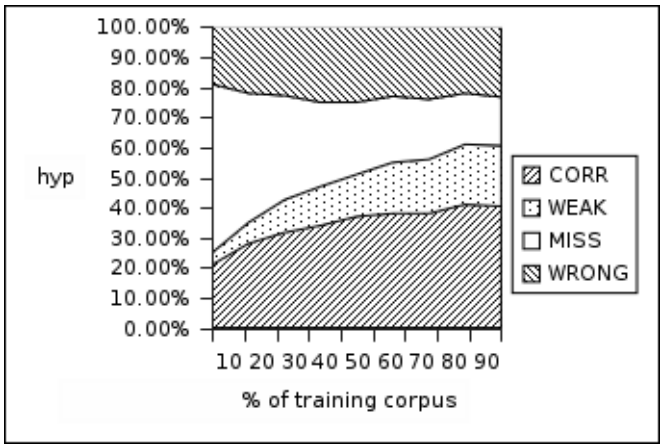

Figure 21: Hypothesis Distribution, c=3, k=4 (*Anabasis*)

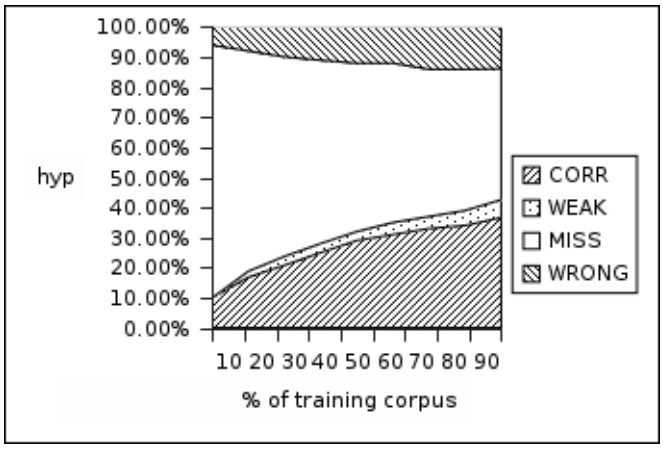

Figure 22: Hypothesis Distribution, c=4, k=4 (*Anabasis*)

In these figures, the four reconstruction hypothesis outcomes described in section are labelled as CORR, for correct hypotheses; MISS, for missing hypotheses; WEAK, for weak hypotheses; and WRONG, for wrong hypotheses.

The data Figures 19 through 22 summarize came from performing 100 reconstructions of 4-holes across each (c, p) combination of input parameters, where c, the amount of context, ranged from 1 to 4 characters, and p, the size of the training text subset, ranged from 10 to 90 percent of *Anabasis* in 10 percent increments, as described previously.

Intepretation

We can now interpret the results shown in section . In brief, the following relationships between training corpus size and reconstruction performance emerge:

1. Within a single figure, which represents a variable quantity of training data, and a fixed amount of context:

   (a) the missing hypothesis fraction decreases as training text increases;

   (b) the weak hypothesis fraction increases as training text increases;

   (c) the correct hypothesis fraction increases moderately with increasing training text.

hole size of 4 was chosen for this study because a large hole with little context is the worst-case scenario for MTR, and thus the case in which the MTR algorithm performs least well.

2. Across the series of figures, which represent a varying amount of available reconstruction context, the correct and missing hypothesis fractions grow as available context increases.[3]

The MTR algorithm outlined in section is in fact very conservative in the way it forms reconstruction hypotheses in the final combined hypothesis set: a hypothesis may only appear in the final hypothesis set if it was in *all* of the earlier sets of proposed reconstructions. This is a by-product of DEMPSTER'S RULE, which "zeroes out" the probabilities of hypotheses which do not appear in all the constituent hypothesis sets.

Bearing in mind the conservative way the MTR algorithm forms reconstruction hypotheses, that the missing hypothesis fraction (MISS in Figures 19 through 22) would decline with increasing training text makes sense; as training text increases, the set of of available n-grams is enriched, and thus it is easier to secure "consensus" between the multiple initial reconstruction hypothesis sets. The enhanced degree of intersection between these sets manifests as a reduction in missing hypotheses. For a fixed amount of context c, that the weak hypothesis fraction would increase with training text shows this same effect: the correct reconstruction appears in the final combined hypothesis set with greater frequency, though with with sub-maximal probability.

That the correct hypothesis fraction increases modestly with training text comes as little surprise. Viewed in reverse, one would of course expect that a very small amount of training text would yield poor results, in terms of rates of correct reconstructions.

More intriguing is the combined effect of adding context and increasing training text simultaneously. In this case, the total fraction of correct and missing hypotheses combined, increases with the amount of training text supplied. The interesting possibility is that by adopting a modified (and less conservative) mechanism for combining reconstructions hypotheses, we may be able be able to tap into the large pool of missing hypothesis outcomes, and improve the rate at which MTR generates correct reconstructions. For example, a "voting" scheme could be used in which an initial hypothesis

[3]Note, though, that adding more context may *decrease* the proportion of *correct* hypotheses! The implication here is that more context is not necessarily better, unless we take additional measures to augment the original MTR algorithm.

only had to appear in a large fraction of the multiple hypothesis sets created in the mass function generation step of the MTR algorithm, and in this way a "missing hypothesis" outcome could be converted to one of the other types.

Lastly, the rate of correct reconstruction could be improved if there were a way to convert weak reconstructions (which increase with training text) into correct reconstructions. One such means would be to apply additional context, beyond that provided by simple character-oriented n-grams, in the form of syntactic constraints on the text surrounding a k-hole. This type of grammatical constraint propagation has been used successfully in OCR to improve recognition rates [18], and presents a promising avenue for further improvement to the basic MTR algorithm.

APPENDIX C

DEMPSTER'S RULE FOR MTR

APPENDIX C

DEMPSTER'S RULE FOR MTR

This appendix contains a small but fully-worked example of how DEMPSTER'S RULE is used to combine multiple mass functions into a single mass function.

Suppose we have a damaged document similar to Xenophon's *Anabasis*, and thus we use *Anabasis* as a training text for our reconstruction attempt. From *Anabasis* we construct n-gram tables which represent the language used by the damaged source document.

Suppose the damaged document contains a 1-hole, such as

```
WITH C?RUS TO
```

with the "?" symbol representing the 1-hole.

Using 2 characters of context, we can slide a 3-gram window across the hole, yielding the specific patterns ⟨ C?⟩, ⟨C?R⟩, and ⟨?RU⟩.

Tables 21, 22, and 23 give the reconstruction probabilities for the specific patterns ⟨ C?⟩, ⟨C?R⟩, and ⟨?RU⟩, respectively. The n-grams listed in the tables are constructed by inserting the hypothesized symbol into the specific pattern used for each table, and the frequencies given represent the absolute frequencies of these n-grams as they occur in the training text.

The reconstructions for specific pattern ⟨ C?⟩ are {A, H, L, O, Y}. The mass function m_1 for this specific pattern is the set of tuples (r, p) where r is a reconstruction and p is the associated

hypothesis h	3-gram	frequency f_h	probability $p_h = f_h/S$
{A}	{ CA}	700	0.28
{H}	{ CH}	300	0.12
{L}	{ CL}	200	0.08
{O}	{ CO}	1100	0.44
{Y}	{ CY}	200	0.08
	$S = \sum f_h = 2500$		

Table 21: Reconstruction Probabilities: Specific Pattern ⟨ C?⟩

hypothesis h	3-gram	frequency f_h	probability $p_h = f_h/S$
{A}	{CAR}	100	0.1818
{E}	{CER}	150	0.2727
{O}	{COR}	100	0.1818
{Y}	{CYR}	200	0.3636
	$S = \sum f_h = 550$		

Table 22: Reconstruction Probabilities: Specific Pattern ⟨C?R⟩

reconstruction probability. Thus,

$$m_1 = \{\langle \mathtt{A}, 0.28\rangle, \langle \mathtt{H}, 0.12\rangle, \langle \mathtt{L}, 0.08\rangle, \langle \mathtt{O}, 0.44\rangle, \langle \mathtt{Y}, 0.08\rangle\}$$

The reconstructions for specific pattern ⟨C?R⟩ are {A, E, O, Y}. The mass function for this pattern is

$$m_2 = \{\langle \mathtt{A}, 0.182\rangle, \langle \mathtt{E}, 0.273\rangle, \langle \mathtt{O}, 0.182\rangle, \langle \mathtt{Y}, 0.363\rangle\}$$

The reconstructions a in the intersection of these two hypothesis sets is {A, O, Y}. For these reconstructions, we calculate their probabilities using DEMPSTER'S RULE.

$$(m_1 \oplus m_2)(a) = \frac{\sum m_1(x)m_2(y)\ \forall x, y \ni x \cap y = \{a\}}{1 - \sum m_1(x)m_2(y)\ \forall x, y \ni x \cap y = \emptyset}$$

Note though, that for purposes of MTR a single hypothesis a is a singleton set.[1] This enables us to simplify the numerator of DEMPSTER'S RULE by removing the summation, yielding just

$$(m_1 \oplus m_2)(a) = \frac{m_1(a)m_2(a)}{1 - \sum m_1(x)m_2(y)\ \forall x, y \ni x \cap y = \emptyset}$$

We will tackle this calculation in two parts: the numerators N_a and the denominator D of the rule. The numerator calculation follows:

[1] This is a consequence of the way we constructed hypotheses from n-grams. The summation in the numerator of Dempster's Rule in its original form handles the case where the intersections of mass functions have multiple elements. Fortunately, this generality is not required for MTR.

hypothesis h	3-gram	frequency f_h	probability $p_h = f_h/S$
{A}	{ARU}	5	0.02
{O}	{ORU}	5	0.02
{T}	{TRU}	100	0.32
{Y}	{YRU}	200	0.64
	$S = \sum f_h = 310$		

Table 23: Reconstruction Probabilities: Specific Pattern ⟨?RU⟩

a	$m_1(a)$	$m_2(a)$	$N_a = m_1(a)m_2(a)$
{A}	0.28	0.182	0.0510
{O}	0.44	0.182	0.0801
{Y}	0.08	0.363	0.0290

Next, we calculate the denominator D for $a \in \{\texttt{A}, \texttt{O}, \texttt{Y}\}$: Note that the all the denominators are identical for each of the hypotheses {A, O, Y} in the intersection of mass functions m_1 and m_2, and thus we only need to calculate this denominator only once.

x	y	$m_1(x)$	$m_2(y)$	$T_a = m_1(x)m_2(y)$
{A}	{E}	0.28	0.273	0.0764
{A}	{O}	0.28	0.182	0.0510
{A}	{Y}	0.28	0.363	0.1016
{H}	{A}	0.12	0.182	0.0218
{H}	{E}	0.12	0.273	0.0328
{H}	{O}	0.12	0.182	0.0218
{H}	{Y}	0.12	0.363	0.0436
{L}	{A}	0.08	0.182	0.0146
{L}	{E}	0.08	0.273	0.0218
{L}	{O}	0.08	0.182	0.0146
{L}	{Y}	0.08	0.363	0.0290
{O}	{A}	0.44	0.182	0.0801
{O}	{E}	0.44	0.273	0.1201
{O}	{Y}	0.44	0.363	0.1597
{Y}	{A}	0.08	0.182	0.0146
{Y}	{E}	0.08	0.273	0.0218
{Y}	{O}	0.08	0.182	0.0146

$$\sum T_a = 0.8399$$
$$D = 1 - T_a = 0.1601$$

Finally, we can divide the numerators N_a by the DEMPSTER'S RULE denominator D to calculate the probabilities in the combined mass function $(m_1 \oplus m_2)$.

a	N_a	D	$N_a/D = (m_1 \oplus m_2)(a)$
{A}	0.0510	0.1601	.3186
{O}	0.0801	0.1601	.5003
{Y}	0.0290	0.1601	.1811
			$\sum = 1$

This combined mass function $(m_1 \oplus m_2)$, hereafter written $m_{1\oplus 2}$, as constructed from mass functions m_1 and m_2 with probabilities rounded to 2 decimal places, is

$$m_{1\oplus 2} = \{\langle \texttt{A}, 0.32\rangle, \langle \texttt{O}, 0.50\rangle, \langle \texttt{Y}, 0.18\rangle\}$$

This mass function represents the combined reconstruction probabilities obtained from the mass functions associated with the specific patterns $\langle$ C?$\rangle$ and $\langle$C?R$\rangle$. For the third and final specific pattern $\langle$?RU$\rangle$, we have mass function

$$m_3 = \{\langle \mathtt{A}, 0.02\rangle, \langle \mathtt{O}, 0.02\rangle, \langle \mathtt{T}, 0.32\rangle, \langle \mathtt{Y}, 0.64\rangle\}$$

which can be combined with $m_{1\oplus 2}$ using another application of DEMPSTER'S RULE. When combining m_3 with $m_{1\oplus 2}$, the intersection of the hypotheses from these two mass functions is again just the set {A,O,Y}. The numerator calculation for this application of DEMPSTER'S RULE follows:

a	$m_3(a)$	$m_{1\oplus 2}(a)$	$N_a = m_3(a)m_{1\oplus 2}(a)$
{A}	0.02	0.32	0.0064
{O}	0.02	0.50	0.0100
{Y}	0.64	0.18	0.1152

The denominator calculation follows the method shown previously:

x	y	$m_3(x)$	$m_{1\oplus 2}(y)$	$T_a = m_3(x)m_{1\oplus 2}(y)$
{A}	{O}	0.02	0.50	0.0100
{A}	{Y}	0.28	0.18	0.0036
{O}	{A}	0.44	0.32	0.0064
{O}	{Y}	0.44	0.18	0.0036
{T}	{A}	0.08	0.32	0.1024
{T}	{O}	0.08	0.50	0.1600
{T}	{Y}	0.08	0.18	0.0576
{Y}	{A}	0.08	0.32	0.2048
{Y}	{O}	0.08	0.50	0.3200

$$\sum T_a = 0.8684$$
$$D = 1 - T_a = 0.1316$$

To calculate the probabilities for mass function $m_{1\oplus 2\oplus 3}$, we continue:

a	N_a	D	$N_a/D = m_{1\oplus 2\oplus 3}(a)$
{A}	0.0064	0.1316	.0486
{O}	0.0100	0.1316	.0760
{Y}	0.1152	0.1316	.8754
			$\sum = 1$

Thus, the final mass function constructed from mass functions m_1, m_2, and m_3, with probabilities rounded to 2 decimal places, is

$$m_{1\oplus 2\oplus 3} = \{\langle \mathtt{A}, 0.05\rangle, \langle \mathtt{O}, 0.08\rangle, \langle \mathtt{Y}, 0.87\rangle\}$$

The final stage of the MTR algorithm would select from mass function $m_{1\oplus 2\oplus 3}$ the reconstruction with the highest probability as the "best" reconstruction. Thus, the reconstruction $\langle \texttt{Y} \rangle$ would be chosen, and this in turn would yield the reconstructed fragment

```
WITH CYRUS TO
```

when given the damaged text fragment given previously.

APPENDIX D

ALTERNATIVE PROBABILITY COMBINING RULES FOR MTR

APPENDIX D

ALTERNATIVE PROBABILITY COMBINING RULES FOR MTR

The AND RULE and OR RULE are probability combining rules examined as alternatives to DEMPSTER'S RULE. These two rules are frequentistic reinterpretations from the opposite ends of the spectrum of combining rules discussed in [36] and [45]. Although these rules are simpler than DEMPSTER'S RULE, they nonetheless exhibit interesting properties: both rules are computationally much less burdensome than DEMPSTER'S RULE; the AND RULE only performs slightly less well than DEMPSTER'S RULE, in terms of mean rates of correct reconstruction; and the OR RULE outperforms DEMPSTER'S RULE in terms of minimizing the mean rate of wrong reconstructions.

Details of these alternative rules follow, along with worked examples and the results of substituting these rules for DEMPSTER'S RULE on the three corpora studied.

The AND RULE

Let M be the set of mass functions we want to combine:

$$M = \{m_1, m_2, ..., m_j\}$$

and H_M be the set of hypotheses common to all the mass functions m_i.

$$H_M = \bigcap_{i=1}^{j} \text{hyp}(m_i) = \{b_1, b_2, ..., b_r\}$$

Let

$$f(x, m_i) = \text{abs. freq. of hypothesis } x \text{ in } m_i$$

then the AND RULE lets us compute new mass function probabilities C so that

$$C(a, M) = \frac{\sum_{i=1}^{j} f(a, m_i)}{\sum_{q=1}^{r} \sum_{i=1}^{j} f(b_q, m_i)} \text{ where } a, b_q \in H_M.$$

$m_1 = \{\ \langle \texttt{A}, 700\rangle,\ \langle \texttt{H}, 300\rangle,\ \langle \texttt{L}, 200\rangle,\ \langle \texttt{O}, 1100\rangle,\ \langle \texttt{Y}, 200\rangle\ \}$
$m_2 = \{\ \langle \texttt{A}, 100\rangle,\ \langle \texttt{E}, 150\rangle,\ \langle \texttt{O}, 100\rangle,\ \langle \texttt{Y}, 200\rangle\ \}$
$m_3 = \{\ \langle \texttt{A}, 5\rangle,\ \langle \texttt{O}, 5\rangle,\ \langle \texttt{T}, 100\rangle,\ \langle \texttt{Y}, 200\rangle\ \}$

Table 24: Mass Functions for AND RULE Example

Example Calculation using the AND RULE

To illustrate how the AND RULE works, a fully-worked example is now provided, using the three example reconstruction mass functions given in Appendix C. The three mass functions (m_1, m_2, and m_3) are presented in Table 24 as sets of 2-tuples. Within a tuple, the first element is a reconstruction hypothesis, and the second element is the absolute frequency of the corresponding n-grams observed in a training text.

H_M, the set of hypotheses common to m_1, m_2, and m_3, is the set $\{\texttt{A}, \texttt{O}, \texttt{Y}\}$. Applying the AND RULE we obtain

$$\begin{aligned}
C(\{\texttt{A}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{A}\}, m_i)}{\sum_{q=1}^{2}\sum_{i=1}^{3} f(b_q, m_i)} \\
&= \frac{700+100+5}{(700+100+5)+(1100+100+5)+(200+200+200)} \\
&= 805/(805+1205+600) \\
&= 0.308 \\
C(\{\texttt{O}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{O}\}, m_i)}{\sum_{q=1}^{2}\sum_{i=1}^{3} f(b_q, m_i)} \\
&= \frac{1100+100+5}{(700+100+5)+(1100+100+5)+(200+200+200)} \\
&= 1205/(805+1205+600) \\
&= 0.462 \\
C(\{\texttt{Y}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{Y}\}, m_i)}{\sum_{q=1}^{2}\sum_{i=1}^{3} f(b_q, m_i)} \\
&= \frac{200+200+200}{(700+1100+5)+(1100+100+5)+(200+200+200)} \\
&= 600/(805+1205+600) \\
&= 0.230
\end{aligned}$$

which yields a new mass function $\{\langle \texttt{A}, 0.308\rangle,\ \langle \texttt{O}, 0.462\rangle,\ \langle \texttt{Y}, 0.249\rangle\}$.

Two computational characteristics of the AND RULE are worthy of note. In the AND RULE the component mass functions being combined are all combined in a single step, as opposed to accumulating the combined mass function in pairwise-fashion, as is done with DEMPSTER'S RULE.

Secondly, in DEMPSTER'S RULE, for every pairwise combination of two mass functions into one, there is an $O(n^2)$ summation[1]; in the AND RULE, the double summation in the denominator of the combining rule

$$\sum_{q=1}^{r}\sum_{i=1}^{j} f(b_q, m_i) \text{ where } a, b_q \in H_M.$$

reduces to a linear-time calculation that is only performed once. The reason for this speed-up in the AND RULE is that the number of mass functions being combined for a given hole is small and, in practice, bounded by a constant. More precisely, the number of distinct mass functions to combine is $c + 1$, where c is the number of characters of context used for the reconstruction attempt. Since combining two mass functions can be performed in linear time, and there is a constant number of mass functions to combine, the overall calculation runs in linear time.

[1]Here, $O(n^2)$ is really $O(xy)$, where x and y are the sizes of the two hypothesis sets; as the two sets tend to have similar sizes, the summation is effectively quadratic in worst-case time complexity.

context c	hole size k	% correct	% weak	% missing	% wrong
1	1	31	69	0	0
2	1	54	46	0	0
3	1	79	19	1	1
4	1	84	5	10	2
1	2	18	82	0	0
2	2	41	57	0	2
3	2	67	24	4	5
4	2	66	7	21	6
1	3	11	88	0	1
2	3	30	62	0	7
3	3	52	25	10	13
4	3	51	8	32	9
1	4	6	89	0	5
2	4	22	59	1	19
3	4	40	24	14	22
4	4	38	7	42	13

Table 25: MTR Results, AND RULE, *Anabasis*

AND RULE Results

Tables 25, 26, and 27 present the AND RULE missing-text reconstruction rates, for the three test corpora (*Anabasis*, *Moby Dick*, *Vulgate*), respectively. Results are given for the four reconstruction outcomes (correct, weak, missing, and wrong reconstructions) discussed previously in Section . The experimental method and reconstructions attempted were identical to those described in Chapter 3, except that the AND RULE was substituted for DEMPSTER'S RULE as the mass function combining rule.

For visual perspective, the same data in Tables 25, 26, and 27 are presented in Figures 23 through 26 for *Anabasis*, Figures 27 through 30 for *Moby Dick*, and Figures 31 through 34 for *Vulgate*.

Interpretation

Table 29 provides a digest of data from Tables 5 and 28, and compares the percentage rates of correct missing-text reconstruction for DEMPSTER'S RULE and the AND RULE, in side-by-side fashion, for the three test corpora.

The means of the differences shown at the bottom of Table 29 indicate that the AND RULE

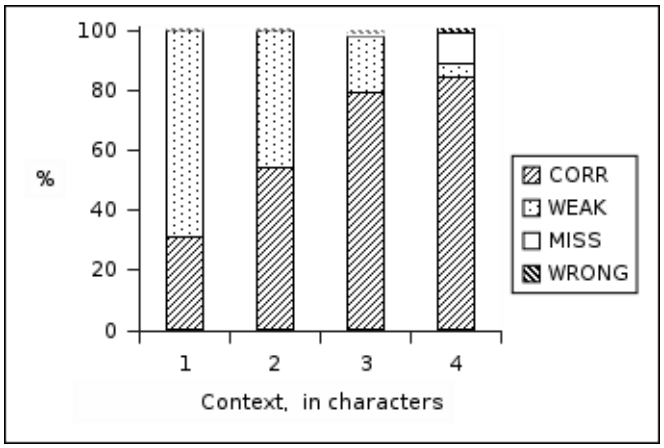

Figure 23: Context vs. MTR Performance (hole size = 1, AND RULE, *Anabasis*)

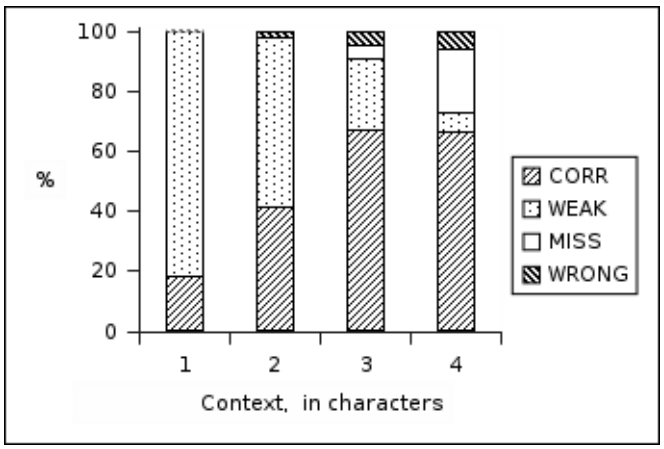

Figure 24: Context vs. MTR Performance (hole size = 2, AND RULE, *Anabasis*)

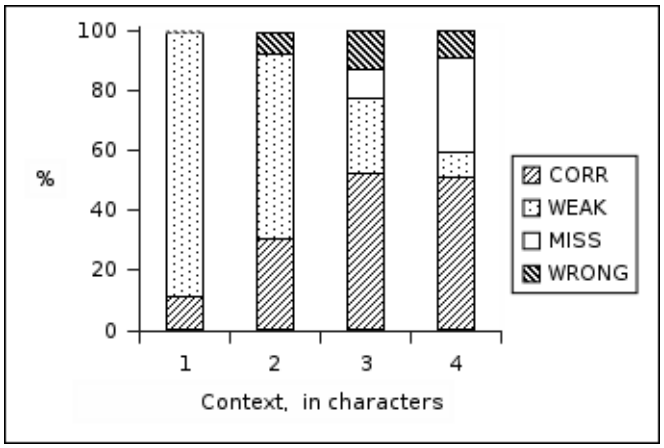

Figure 25: Context vs. MTR Performance (hole size = 3, AND RULE, *Anabasis*)

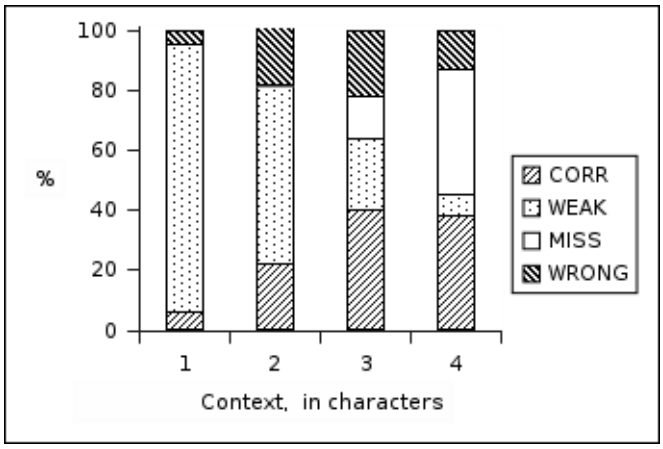

Figure 26: Context vs. MTR Performance (hole size = 4, AND RULE, *Anabasis*)

context c	hole size k	% correct	% weak	% missing	% wrong
1	1	29	71	0	0
2	1	52	48	0	0
3	1	73	25	1	1
4	1	85	8	6	2
1	2	16	84	0	0
2	2	35	64	0	1
3	2	61	32	2	5
4	2	64	12	16	7
1	3	9	90	0	1
2	3	23	72	0	4
3	3	44	37	4	15
4	3	45	12	28	15
1	4	5	91	0	4
2	4	16	68	0	16
3	4	31	30	10	29
4	4	31	10	37	21

Table 26: MTR Results, AND RULE, *Moby Dick*

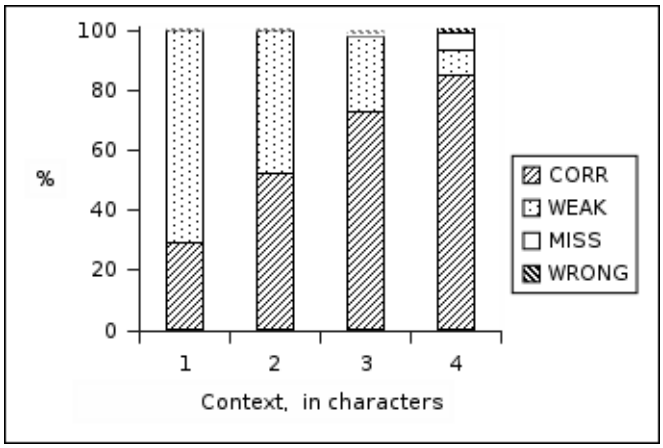

Figure 27: Context vs. MTR Performance (hole size = 1, AND RULE, *Moby Dick*)

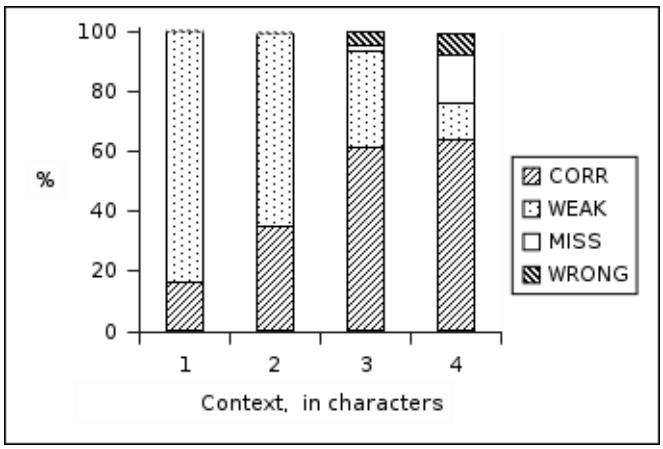

Figure 28: Context vs. MTR Performance (hole size = 2, AND RULE, *Moby Dick*)

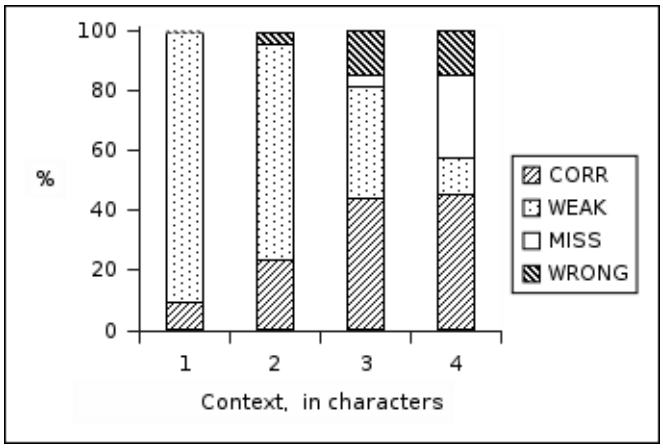

Figure 29: Context vs. MTR Performance (hole size = 3, AND RULE, *Moby Dick*)

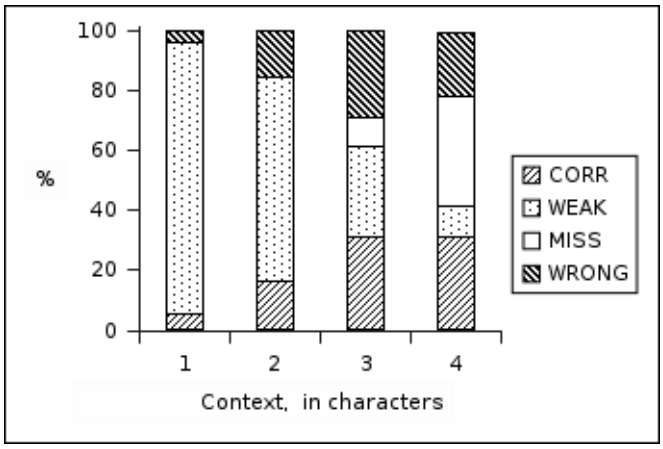

Figure 30: Context vs. MTR Performance (hole size = 4, AND RULE, *Moby Dick*)

context c	hole size k	% correct	% weak	% missing	% wrong
1	1	28	72	0	0
2	1	48	52	0	0
3	1	71	29	0	0
4	1	85	13	2	1
1	2	11	89	0	0
2	2	33	67	0	0
3	2	59	39	0	2
4	2	74	20	3	3
1	3	8	92	0	0
2	3	22	77	0	1
3	3	44	50	1	5
4	3	59	23	8	10
1	4	4	96	0	1
2	4	14	80	0	6
3	4	34	48	2	16
4	4	46	23	13	18

Table 27: MTR Results, AND RULE, *Vulgate*

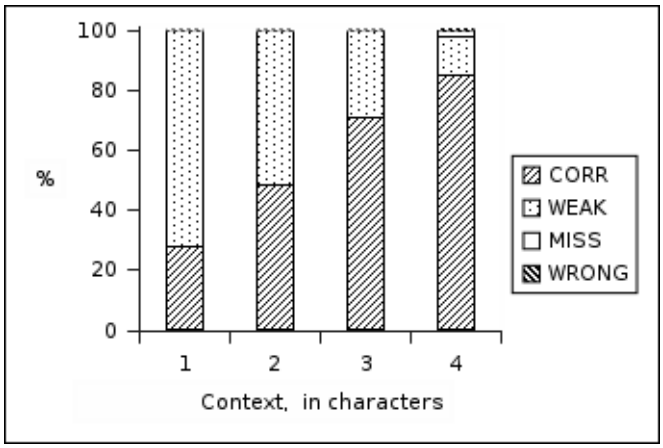

Figure 31: Context vs. MTR Performance (hole size = 1, AND RULE, *Vulgate*)

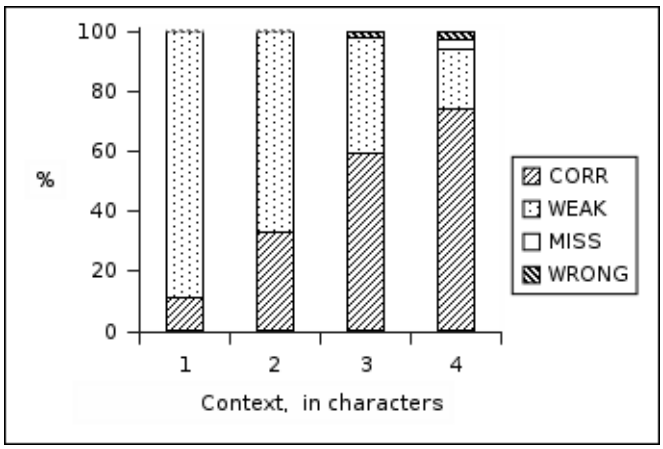

Figure 32: Context vs. MTR Performance (hole size = 2, AND RULE, *Vulgate*)

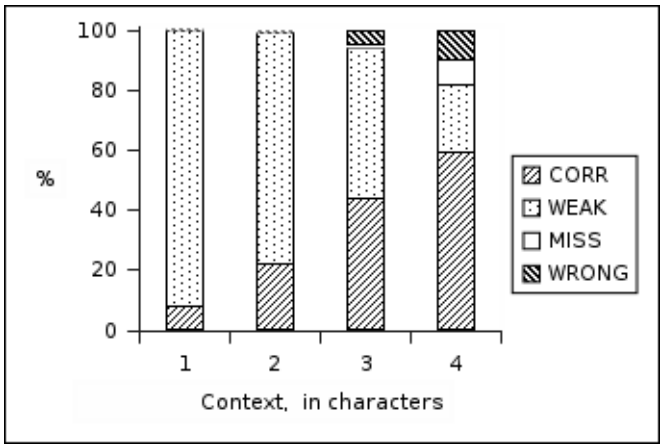

Figure 33: Context vs. MTR Performance (hole size = 3, AND RULE, *Vulgate*)

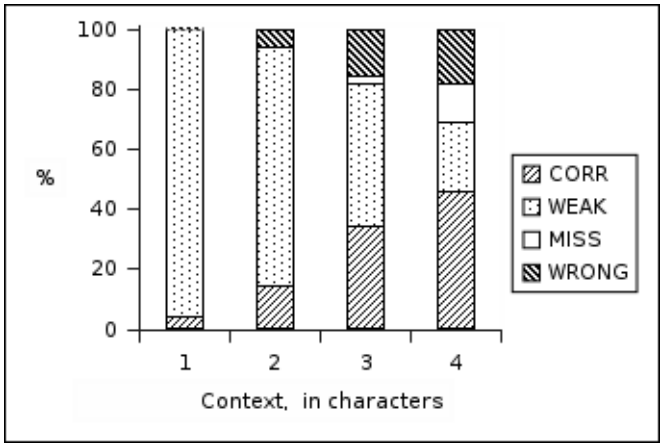

Figure 34: Context vs. MTR Performance (hole size = 4, AND RULE, *Vulgate*)

context c	hole size k	% correct, *Anabasis*	% correct, *Moby Dick*	% correct, *Vulgate*
1	1	31	29	28
2	1	54	52	48
3	1	79	73	71
4	1	84	85	85
1	2	18	16	11
2	2	41	35	33
3	2	67	61	59
4	2	66	64	74
1	3	11	9	8
2	3	30	23	22
3	3	52	44	44
4	3	51	45	59
1	4	6	5	4
2	4	22	16	14
3	4	40	31	34
4	4	38	31	48
corpus size, in kilobytes		461	1138	4050

Table 28: Rates of Correct Reconstruction for Three Corpora, AND RULE

		Anabasis			*Moby Dick*			*Vulgate*		
context	hole	DEMP	AND	△	DEMP	AND	△	DEMP	AND	△
1	1	39	31	8	37	29	8	31	28	3
2	1	63	54	9	61	52	9	57	48	9
3	1	84	79	5	80	73	7	79	71	8
4	1	84	84	0	86	85	1	89	85	4
1	2	23	18	5	20	16	4	16	11	5
2	2	48	41	7	44	35	9	42	33	9
3	2	70	67	3	66	61	5	68	59	9
4	2	67	66	1	66	64	2	77	74	3
1	3	14	11	3	12	9	3	9	8	1
2	3	37	30	7	30	23	7	30	22	8
3	3	55	52	3	49	44	5	50	44	6
4	3	51	51	0	46	45	1	62	59	3
1	4	7	6	1	6	5	1	5	4	1
2	4	27	22	5	20	16	4	20	14	6
3	4	42	40	2	33	31	2	39	34	5
4	4	38	38	0	32	31	1	48	46	2
mean △				3.7			4.3			5.1

Table 29: Comparison of Rates of Correct Reconstruction, DEMPSTER'S RULE vs. AND RULE

generally performs less well than DEMPSTER'S RULE, by a margin of approximately 3 to 5 percent, in terms of correct reconstructions achieved. On the other hand, as previously noted in section , the AND RULE is algorithmically superior to DEMPSTER'S RULE in terms of worst-case time complexity ($O(n)$ vs. $O(n^2)$).

The OR RULE

Let M be the set of mass functions we want to combine:

$$M = \{m_1, m_2, ..., m_j\}$$

and H_M be the union of all the hypotheses in the mass functions m_i

$$H_M = \bigcup_{i=1}^{j} \text{hyp}(m_i) = \{b_1, b_2, ..., b_r\}$$

Let

$$f(x, m_i) = \text{abs. freq. of hypothesis } x \text{ in } m_i$$

and further, if hypothesis $b \notin$ mass function m, let $f(b, m) = 0$. With this constraint, we can define

$m_1 = \{\ \langle \texttt{A}, 700\rangle,\ \langle \texttt{H}, 300\rangle,\ \langle \texttt{L}, 200\rangle,\ \langle \texttt{O}, 1100\rangle,\ \langle \texttt{Y}, 200\rangle\ \}$
$m_2 = \{\ \langle \texttt{A}, 100\rangle,\ \langle \texttt{E}, 150\rangle,\ \langle \texttt{O}, 100\rangle,\ \langle \texttt{Y}, 200\rangle\ \}$
$m_3 = \{\ \langle \texttt{A}, 5\rangle,\ \langle \texttt{O}, 5\rangle,\ \langle \texttt{T}, 100\rangle,\ \langle \texttt{Y}, 200\rangle\ \}$

Table 30: Mass Functions for OR RULE Example

the OR RULE which computes a new mass function probability C as

$$C(a, M) = \frac{\sum_{i=1}^{j} f(a, m_i)}{\sum_{q=1}^{r} \sum_{i=1}^{j} f(b_q, m_i)} \text{ where } a, b_q \in H_M.$$

Example Calculation using the OR RULE

To illustrate how the OR RULE works, a fully-worked example is now provided, using the three example reconstruction mass functions given in Appendix C. The three mass functions are presented as sets of 2-tuples, the first element of which is a reconstruction hypothesis, and the second element of which is the absolute frequency of the corresponding 3-grams observed in a training text.

To illustrate how the OR RULE works, a fully-worked example is now provided, using the three example reconstruction mass functions given in Appendix C. The three mass functions (m_1, m_2, and m_3) are presented in Table 30 as sets of 2-tuples. Within a tuple, the first element is a reconstruction hypothesis, and the second element is the absolute frequency of the corresponding n-grams observed in a training text.

H_M, the union of all the hypotheses in m_1, m_2, and m_3, is the set $\{\texttt{A}, \texttt{E}, \texttt{H}, \texttt{L}, \texttt{O}, \texttt{T}, \texttt{Y}\}$. Calculating the numerators for the individual hypotheses separately, we obtain

$$\begin{aligned}
\textstyle\sum_{i=1}^{3} f(\{\texttt{A}\}, m_i) &= 100 + 100 + 5 \\
&= 205 \\
\textstyle\sum_{i=1}^{3} f(\{\texttt{E}\}, m_i) &= 0 + 150 + 0 \\
&= 150 \\
\textstyle\sum_{i=1}^{3} f(\{\texttt{H}\}, m_i) &= 300 + 0 + 0 \\
&= 300 \\
\textstyle\sum_{i=1}^{3} f(\{\texttt{L}\}, m_i) &= 200 + 0 + 0 \\
&= 200 \\
\textstyle\sum_{i=1}^{3} f(\{\texttt{O}\}, m_i) &= 1100 + 100 + 5 \\
&= 1205 \\
\textstyle\sum_{i=1}^{3} f(\{\texttt{T}\}, m_i) &= 0 + 0 + 100 \\
&= 100 \\
\textstyle\sum_{i=1}^{3} f(\{\texttt{Y}\}, m_i) &= 200 + 200 + 200 \\
&= 600
\end{aligned}$$

while the denominator is just

$$\begin{aligned}\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i) &= 205 + 150 + 300 + 200 + 1205 + 100 + 600 \\ &= 2760\end{aligned}$$

which enables us now to calculate

$$\begin{aligned}C(\{\texttt{A}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{A}\}, m_i)}{\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{205}{2760} \\ &= 0.074\end{aligned}$$

$$\begin{aligned}C(\{\texttt{E}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{E}\}, m_i)}{\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{150}{2760} \\ &= 0.054\end{aligned}$$

$$\begin{aligned}C(\{\texttt{H}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{H}\}, m_i)}{\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{300}{2760} \\ &= 0.109\end{aligned}$$

$$\begin{aligned}C(\{\texttt{L}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{L}\}, m_i)}{\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{200}{2760} \\ &= 0.072\end{aligned}$$

$$\begin{aligned}C(\{\texttt{O}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{O}\}, m_i)}{\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{1205}{2760} \\ &= 0.437\end{aligned}$$

$$\begin{aligned}C(\{\texttt{T}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{T}\}, m_i)}{\sum_{q=1}^{7}\sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{100}{2760} \\ &= 0.036\end{aligned}$$

context c	hole size k	% correct	% weak	% missing	% wrong
1	1	31	69	0	0
2	1	49	52	0	0
3	1	65	35	0	0
4	1	78	22	0	0
1	2	16	84	0	0
2	2	32	68	0	0
3	2	46	53	0	0
4	2	57	42	0	1
1	3	9	90	0	0
2	3	17	82	0	0
3	3	30	68	0	1
4	3	41	56	0	3
1	4	5	94	0	1
2	4	10	87	0	2
3	4	19	77	0	4
4	4	31	62	0	8

Table 31: MTR Results, OR RULE, *Anabasis*

$$\begin{aligned} C(\{\texttt{Y}\}, M) &= \frac{\sum_{i=1}^{3} f(\{\texttt{Y}\}, m_i)}{\sum_{q=1}^{7} \sum_{i=1}^{3} f(b_q, m_i)} \\ &= \frac{600}{2760} \\ &= 0.217 \end{aligned}$$

which yields a new mass function $\{\langle \texttt{A}, 0.074\rangle, \langle \texttt{E}, 0.054\rangle, \langle \texttt{H}, 0.109\rangle, \langle \texttt{L}, 0.072\rangle, \langle \texttt{O}, 0.437\rangle, \langle \texttt{T}, 0.036\rangle, \langle \texttt{Y}, 0.217\rangle\}$.

In terms of computational complexity, the OR RULE runs in linear time proportional to the size of the union of the hypothesis sets used.

OR RULE Results

Tables 31, 32, and 33 present the OR RULE missing-text reconstruction rates, for the three test corpora (*Anabasis*, *Moby Dick*, *Vulgate*), respectively. Results are given for to the four reconstruction outcomes (correct, weak, missing, and wrong reconstructions) discussed previously in Section . The experimental method and reconstructions attempted were identical to those described in Chapter 3, except that the OR RULE was substituted for DEMPSTER'S RULE as the mass function combining rule.

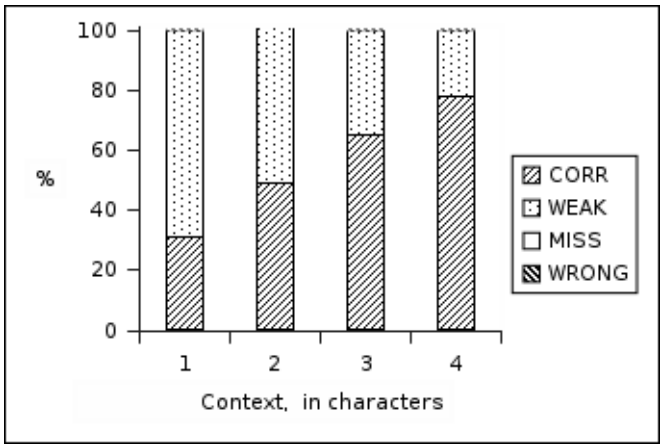

Figure 35: Context vs. MTR Performance (hole size = 1, OR RULE, *Anabasis*)

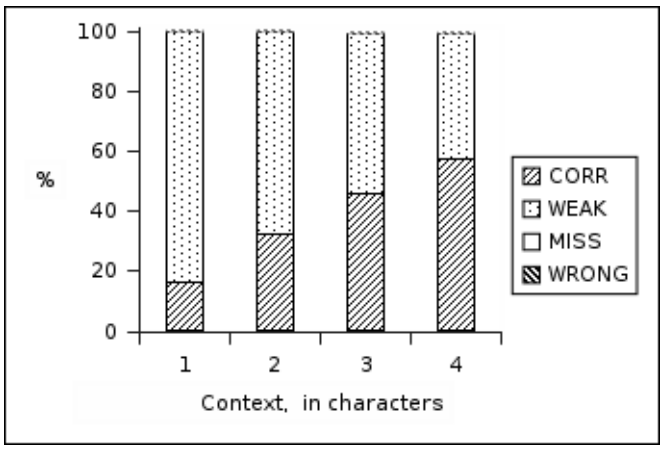

Figure 36: Context vs. MTR Performance (hole size = 2, OR RULE, *Anabasis*)

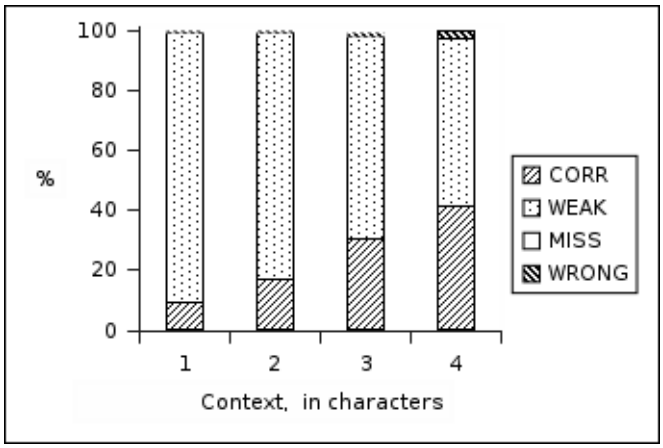

Figure 37: Context vs. MTR Performance (hole size = 3, OR RULE, *Anabasis*)

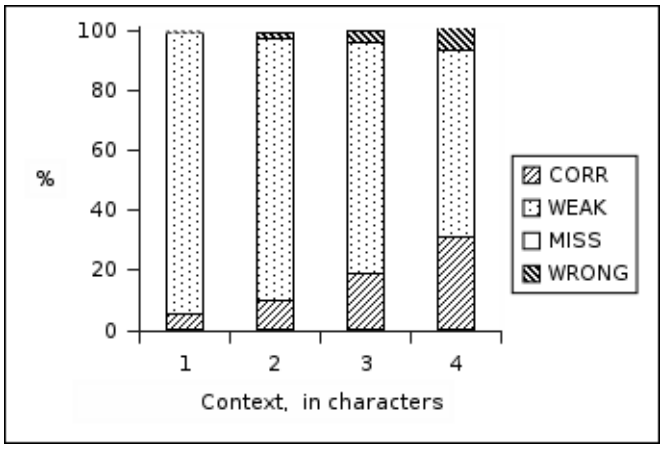

Figure 38: Context vs. MTR Performance (hole size = 4, OR RULE, *Anabasis*)

context c	hole size k	% correct	% weak	% missing	% wrong
1	1	29	71	0	0
2	1	49	51	0	0
3	1	63	37	0	0
4	1	75	25	0	0
1	2	15	85	0	0
2	2	28	72	0	0
3	2	41	59	0	0
4	2	53	47	0	0
1	3	8	92	0	0
2	3	14	86	0	0
3	3	24	76	0	1
4	3	35	63	0	2
1	4	4	95	0	1
2	4	9	89	0	2
3	4	15	82	0	3
4	4	25	68	0	7

Table 32: MTR Results, OR RULE, *Moby Dick*

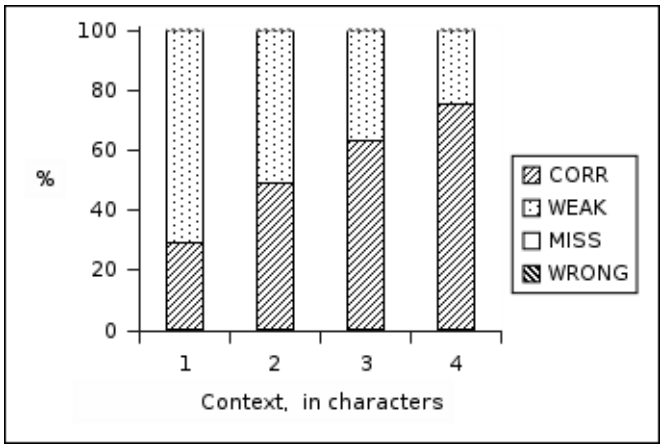

Figure 39: Context vs. MTR Performance (hole size = 1, OR RULE, *Moby Dick*)

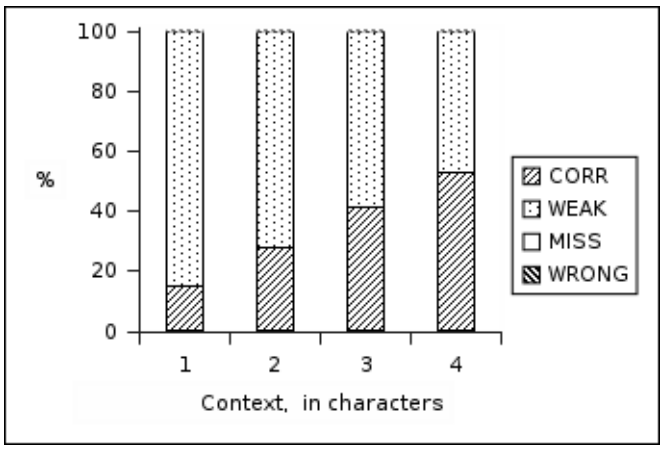

Figure 40: Context vs. MTR Performance (hole size = 2, OR RULE, *Moby Dick*)

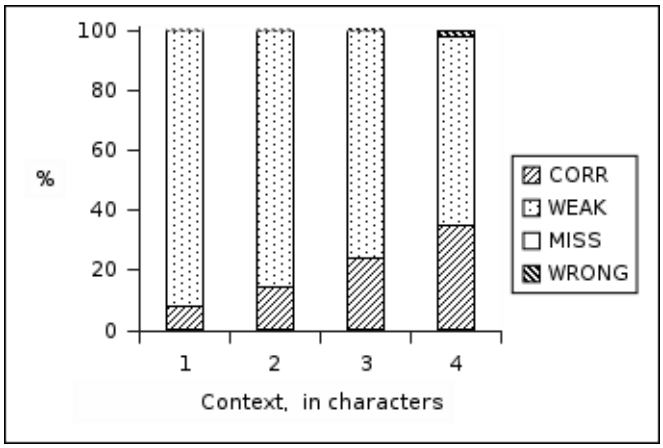

Figure 41: Context vs. MTR Performance (hole size = 3, OR RULE, *Moby Dick*)

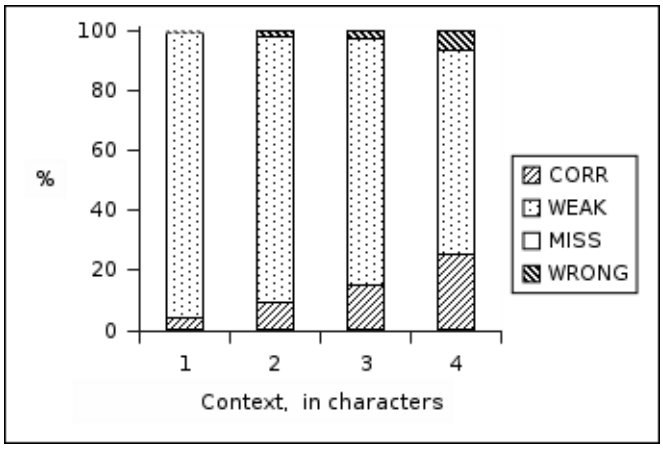

Figure 42: Context vs. MTR Performance (hole size = 4, OR RULE, *Moby Dick*)

context c	hole size k	% correct	% weak	% missing	% wrong
1	1	27	73	0	0
2	1	45	55	0	0
3	1	62	38	0	0
4	1	74	26	0	0
1	2	11	89	0	0
2	2	27	73	0	0
3	2	42	58	0	0
4	2	56	44	0	0
1	3	6	94	0	0
2	3	14	86	0	0
3	3	27	72	0	0
4	3	38	62	0	1
1	4	3	97	0	0
2	4	8	92	0	0
3	4	18	81	0	1
4	4	27	72	0	1

Table 33: MTR Results, OR RULE, *Vulgate*

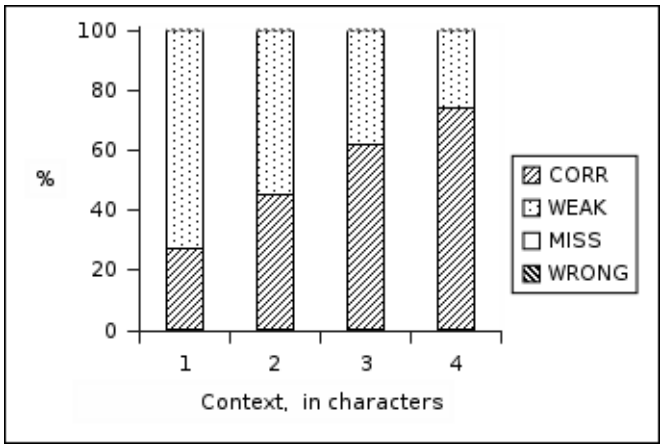

Figure 43: Context vs. MTR Performance (hole size = 1, OR RULE, *Vulgate*)

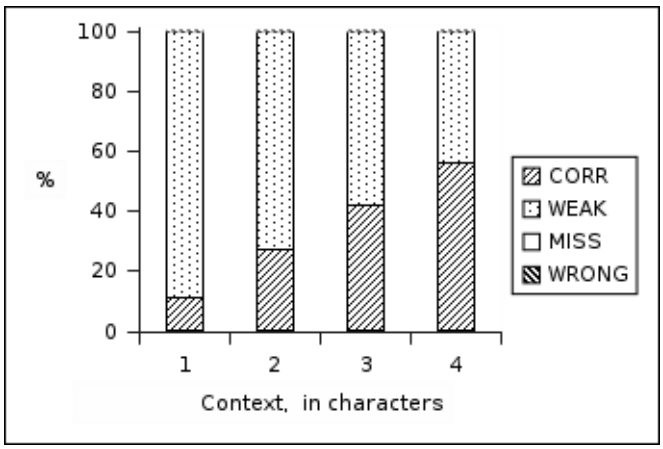

Figure 44: Context vs. MTR Performance (hole size = 2, OR RULE, *Vulgate*)

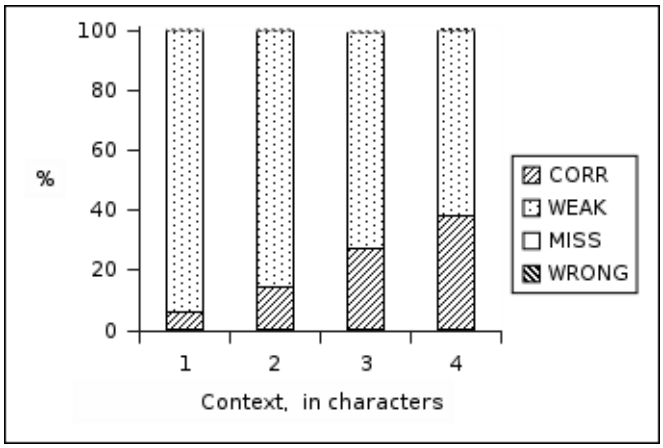

Figure 45: Context vs. MTR Performance (hole size = 3, OR RULE, *Vulgate*)

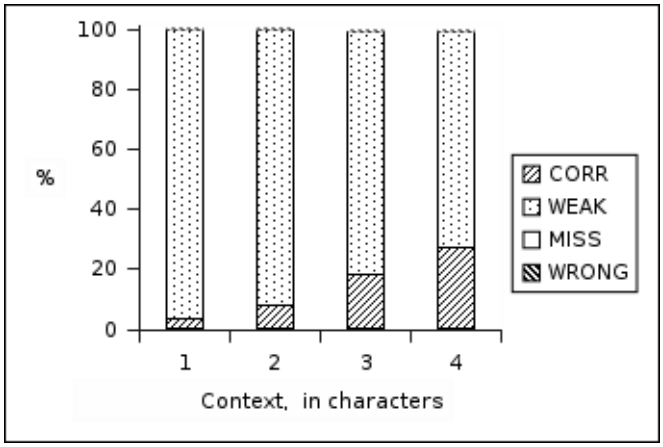

Figure 46: Context vs. MTR Performance (hole size = 4, OR RULE, *Vulgate*)

context c	hole size k	% correct, *Anabasis*	% correct, *Moby Dick*	% correct, *Vulgate*
1	1	31	29	27
2	1	49	49	45
3	1	65	63	62
4	1	78	75	74
1	2	16	15	11
2	2	32	28	27
3	2	46	41	42
4	2	57	53	56
1	3	9	8	6
2	3	17	14	14
3	3	30	24	27
4	3	41	35	38
1	4	5	4	3
2	4	10	9	8
3	4	19	15	18
4	4	31	25	27
corpus size, in kilobytes		461	1138	4050

Table 34: Rates of Correct Reconstruction for Three Corpora, OR RULE

		Anabasis			*Moby Dick*			*Vulgate*		
context	hole	DEMP	OR	△	DEMP	OR	△	DEMP	OR	△
1	1	39	31	8	37	29	8	31	27	4
2	1	63	49	14	61	49	12	57	45	12
3	1	84	65	19	80	63	17	79	62	17
4	1	84	78	6	86	75	11	89	74	15
1	2	23	16	7	20	15	5	16	11	5
2	2	48	32	16	44	28	16	42	27	15
3	2	70	46	24	66	41	25	68	42	26
4	2	67	57	10	66	53	13	77	56	21
1	3	14	9	5	12	8	4	9	6	3
2	3	37	17	20	30	14	16	30	14	16
3	3	55	30	25	49	24	25	50	27	23
4	3	51	41	10	46	35	11	62	38	24
1	4	7	5	2	6	4	2	5	3	2
2	4	27	10	17	20	9	11	20	8	12
3	4	42	19	23	33	15	18	39	18	21
4	4	38	31	7	32	25	7	48	27	21
mean △				13.3			12.6			14.8

Table 35: Comparison of Rates of Correct Reconstruction, DEMPSTER'S RULE vs. OR RULE

For visual perspective, the same data in Tables 31, 32, and 33 are presented in Figures 35 through 38 for *Anabasis*, Figures 39 through 42 for *Moby Dick*, and Figures 43 through 46 for *Vulgate*.

Interpretation

Table 35 provides a digest of data from Tables 5 and 34, and compares the percentage rates of correct missing-text reconstruction for DEMPSTER'S RULE and the OR RULE, in side-by-side fashion, for the three test corpora.

In terms of correct reconstructions achieved, the means of the differences shown at the bottom of Table 35 indicate that the OR RULE performs less well than DEMPSTER'S RULE, by a margin of approximately 13 to 15 percent.

Conversely, Table 36 indicates that the OR RULE performs better than DEMPSTER'S RULE in terms of wrong reconstructions (here, lower rates of wrong reconstruction are better than higher rates). The wrong reconstruction rates of the OR RULE are, on average, from approximately 4 to 6 percent lower than those obtained via DEMPSTER'S RULE. If one examines only the worst-case

		Anabasis			*Moby Dick*			*Vulgate*		
context	hole	DEMP	OR	△	DEMP	OR	△	DEMP	OR	△
1	1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	1	0	1	1	0	1	0	0	0
4	1	2	0	2	2	0	2	1	0	1
1	2	0	0	0	0	0	0	0	0	0
2	2	2	0	2	1	0	1	0	0	0
3	2	5	0	5	5	0	5	2	0	2
4	2	6	1	5	7	0	7	3	0	3
1	3	1	0	1	1	0	1	0	0	0
2	3	7	0	7	4	0	4	1	0	1
3	3	13	1	12	15	1	14	5	0	5
4	3	9	3	6	15	2	13	10	1	9
1	4	5	1	4	4	1	3	1	0	1
2	4	19	2	17	16	2	14	6	0	6
3	4	22	4	18	29	3	26	16	1	15
4	4	13	8	5	21	7	14	18	1	17
mean △				5.3			6.6			3.8

Table 36: Comparison of Rates of Wrong Reconstruction, DEMPSTER'S RULE vs. OR RULE

scenario in MTR[2], the OR RULE outperforms DEMPSTER'S RULE by nearly 10 percent, in terms of obtaining fewer wrong reconstructions.

That the OR RULE should have a remarkably lower rate of wrong reconstructions, is important in that it gives us an improved upper bound on how well MTR can perform. To see why this is so, recall from Section that a wrong reconstruction scenario occurs when the correct reconstruction hypothesis is nowhere to be found in the combined hypothesis set, and hence no possible choice of hypothesis from that set could be correct. That is, any possible choice of hypothesis from a "wrong" hypothesis set is doomed to be incorrect. To minimize the size of the "wrong" hypothesis set, or to make its generation as infrequent as possible, is to increase that likehood that one of the other, more positive, outcomes — those of correct, weak, or missing reconstructions. In two of these latter three outcomes (weak or missing hypotheses) it is in principle possible to apply addition context or methods, beyond the scope of the current study, to obtain a correct reconstruction.

Note that the OR RULE, as defined, can never generate a "missing" hypothesis outcome. This is

[2]MTR works least well when attempting to reconstruct the contents of large holes. In this study, the largest holes used were 4-holes.

because the "missing" outcome is an artifact of rules that are based on logical conjunction – such as DEMPSTER'S RULE or the AND RULE. Specifically, a "missing" outcome occurs when the correct hypothesis is contained in one or more of the base (pre-combination) hypothesis sets, but is not contained in *all* of the pre-combination hypothesis sets. In the OR RULE, this situation cannot arise, since the combining principle is that of logical disjunction: if the correct hypothesis is contained in *any* pre-combination hypothesis set, then that correct hypothesis must appear in the union of all such sets, and a "missing" outcome cannot occur. Experimentally, we can observe the non-occurrence of missing reconstructions for the OR RULE, by examining the "missing" columns of Tables 31 32, and 33.

Zadeh's Anomaly [36] [17] [45] gives us another way of understanding what the presence or absence of "missing" reconstruction outcomes tells us about different types of probability combining rules. The problem Zadeh noted in DEMPSTER'S RULE was that in the presence of strong conflict between the mass functions being combined, DEMPSTER'S RULE may yield counterintuitive results.

To illustrate how this can happen, let us recast the example given in [36, p. 17] in terms of MTR. Suppose we have two mass functions representing different reconstruction hypotheses and their associated probabilities:

$$m_1 = \{\langle \mathtt{A}, 0.99\rangle, \langle \mathtt{Q}, 0.01\rangle\}$$
$$m_2 = \{\langle \mathtt{E}, 0.99\rangle, \langle \mathtt{Q}, 0.01\rangle\}$$

When combining mass functions m_1 and m_2 using DEMPSTER'S RULE, the only reconstruction hypothesis common to both mass functions is $\{\mathtt{Q}\}$; thus, in the combined mass function $m_1 \oplus m_2$, the $\{\mathtt{Q}\}$ hypothesis would have probability 1.0. This result occurs despite m_1 and m_2 both assigning the $\{\mathtt{Q}\}$ hypothesis a very low probability. The basic problem is that a highly improbable result can appear in the combined mass function of a conjunctive combining rule, if the hypothesis was the only hypothesis of nonzero probability common to all the original mass functions to be combined. This outcome can occur in any probability combining rule that is based on logical conjunction, such as DEMPSTER'S RULE and the AND RULE.

REFERENCES CITED

[1] K. Aas and L. Eikvil. Text page recognition using grey-level features and hidden markov models. *Pattern Recognition*, 29(6):997–985, Jun 1996.

[2] David Applebaum. *Probabililty and Information: an integrated approach*. Cambridge University Press, 1996.

[3] C. B. Bose and S. S. Kuo. Connected and degraded text recognition using hidden markov model. *Pattern Recognition*, 27(10):1345–1363, 1994.

[4] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden markov models. *Pattern Recognition*, 28(9):1399–1413, Sep 1995.

[5] S. Calabretto and A. Bozzi. The Philological Workstation BAMBI (Better Access to Manuscripts and Browsing of Images). *Journal of Digital Information*, 1(3), 1998. Available online at URL http://jodi.ecs.soton.ac.uk/, as of 2 July 2003.

[6] Reconstructing Ancient Texts: The Thirty-Seventh Annual Conference on Editorial Problems, November 2-3, 2001, University College, University of Toronto, 2001. Available online at URL http://www.chass.utoronto.ca/papyri/cep/program.html, as of 2 July 2003.

[7] W. Cho, S.-W. Lee, and J.-H. Kim. Modeling and recognition of cursive words with hidden markov models. *Pattern Recognition*, 28(12):1941–1953, Dec 1995.

[8] Jay L. Devore. *Probability and Statistics for Engineering and the Sciences*. Wadsworth, Belmont, CA, 4th edition, 1995.

[9] Yvonne French. Courage and Genius: Tyndale Responsible for Most of King James Translation, Says Biographer. Available online at URL http://www.loc.gov/loc/lcib/9707/daniell.html, as of 11 October 2003, July 1997.

[10] Wendell R. Garner. *Uncertainty and Structure as Psychological Concepts*. Wiley, New York, 1962.

[11] Louis Glassy. Mitre source code. MITRE source code is available online at URL http://isildur.mtech.edu/research/mitre.cpio.Z, as of 10 October 2003, 2003.

[12] Paul Graham. A Plan for Spam. Available online at URL http://www.paulgraham.com/spam.html as of 21 March 2004.

[13] J. J. Hull and S. N. Srihari. Experiments in text recognition with binary n-grams and viterbi algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 4(5):520–530, Sep 1982.

[14] J. J. Hull, S. N. Srihari, and R. Choudhari. An integrated algorithm for text recognition: comparison with a cascaded algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 5(4):384–395, Jul 1983.

[15] Q. Huo and C. Chan. Contextual vector quantization for speech recognition with discrete hidden markov model. *Pattern Recognition*, 28(4):513–517, April 1995.

[16] IDC. Volume of SPAM Doubled in Past Two Years - IDC Expects It Is Only Going to Get Worse. Available online at URL http://www.idc.com/getdoc.jsp?containerId=pr2004_04_12_112824 as of 19 April 2004.

[17] Giorgio Ingargiola. Dempster-Shafer Theory. A brief tutorial on Dempster-Shafer Theory. Available online at URL http://yoda.cis.temple.edu:8080/UGAIWWW/lectures/dempster.html, as of 2 July 2003.

[18] N. Itoh. Japanese language model based on bigrams and its application to on-line character recognition. *Pattern Recognition*, 28(2):135–141, 1995.

[19] Eusebius Hieronymus (St. Jerome). Vulgate Bible. Available online at URL ftp://ftp.std.com/WWW/obi/Religion/Vulgate/, as of 1 July 2003.

[20] David Kahn. *The Codebreakers*. Macmillan, 1967.

[21] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35:400–401, 1987.

[22] H. J. Kim, K. H. Kim, S. K. Kim, and J. K. Lee. On-line recognition of handwritten chinese characters based on hidden markov models. *Pattern Recognition*, 30(9):1489–1500, Sep 1997.

[23] W. S. Kim and R.-H. Park. Offline recognition of handwritten korean and alphanumeric characters using hidden markov model. *Pattern Recognition*, 29(5):845–858, May 1996.

[24] Alan G. Konheim. *Cryptography: a primer*. Wiley, 1981.

[25] Armin Lange. *Computer Aided Text-Reconstruction and Transcription: CATT-Manual*. J. C. B. Mohr, 1993.

[26] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[27] Herman Melville. *Moby Dick*. Project Gutenberg, 2001. The edition used was a machine-readable ASCII copy available online at URL http://www.ibiblio.org/gutenberg/etext01/moby10b.txt, as of 1 July 2003.

[28] Dirk Obbink. Description of "Imaging Papyri" research project. Of particular note is the following statement of research objectives: "The work is very painstaking: the aim is to reconstruct as much as possible of each original papyrus roll and then to reconstruct the text. The reconstruction requires a constant sequence of conjecture, objection, improvement, and eventually, agreement. No single scholar can see all that needs to be done." Available online at URL http://www.classics.ox.ac.uk/research/projects/papyri.asp, as of 28 August 2003.

[29] S.-C. Oh, J.-Y. Ha, and J.-H. Kim. Context-dependent search in interconnected hidden markov model for unconstrained handwriting recognition. *Pattern Recognition*, 28(11):1693–1704, Nov 1995.

[30] H.-S. Park and S.-W. Lee. Offline recognition of large-set handwritten characters with multiple hidden markov models. *Pattern Recognition*, 29(2):231–244, Feb 1996.

[31] David J. Pepper and Mark A. Clements. Phonemic Recognition Using a Large Hidden Markov Model. *IEEE Transactions on Signal Processing*, 40(6):1590–1595, 1992.

[32] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[33] Gordon Raisbeck. *Information Theory: an Introduction for Scientists and Engineers*. MIT Press, 1964.

[34] Neil Rhodes and Jonathan Sawday, editors. *The Renaissance Computer: Knowledge Technology in the First Age of Print*. Routledge, London, 2000.

[35] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, second edition, 2003.

[36] Kari Sentz and Scott Ferson. Combination of Evidence in Dempster-Shafer Theory. Available online at URL http://www.sandia.gov/epistemic/Reports/SAND2002-0835.pdf, as of 25 March 2004.

[37] Glenn Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[38] R. Shinghal and G. T. Toussaint. Experiments in text recognition with the modified viterbi algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1(2):184–193, Apr 1979.

[39] Simon Singh. *The Code Book: the evolution of secrecy from Mary Queen of Scots to quantum cryptography*. Doubleday, 1999.

[40] Abraham Sinkov. *Elementary cryptanalysis: a mathematical approach*. Random House, 1968.

[41] A. R. Smith and L. D. Erdman. Noah: a bottom-up word hypothesizer for large vocabulary speech understanding systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 3(1):41–51, Jan 1981.

[42] Cheng-Huang Tung and Hsi-Jian Lee. Increasing character recognition accuracy by detection and correction of erroneously identified characters. *Pattern Recognition*, 27(9):1259–1266, 1994.

[43] Xenophon. *Anabasis*. Project Gutenberg, 1998. English translation by H. G. Daykns. Available online at URL http://www.ibiblio.org/gutenberg/etext98/anbss10.txt, as of 1 July 2003.

[44] E. J. Yamakoudakis, I. Tsomokos, and P. J. Hutton. n-Grams and their implication to natural language understanding. *Pattern Recognition*, 23(5):509–528, May 1990.

[45] L. A. Zadeh. Review of Books: A Mathematical Theory of Evidence. *AI Magazine*, 5(3):81–83, 1984.