

A New Class of Functions for Describing Logical Structures in Text

by

Ngon D. Dao
B.S. Mechanical Engineering
University of Texas at Austin, 1996


SUBMITTED TO THE
HARVARD-MIT DIVISION OF HEALTH SCIENCES AND TECHNOLOGY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
OF THE DEGREE OF

DOCTOR OF PHILOSOPHY IN MEDICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUNE 2004

© 2004 Ngon Dao. All rights reserved

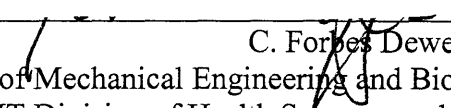
The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic
copies of this thesis document in whole or in part.

Signature of
Author:



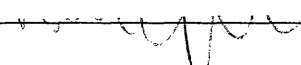
Harvard-MIT Division of Health Sciences and Technology
April 2004

Certified by:

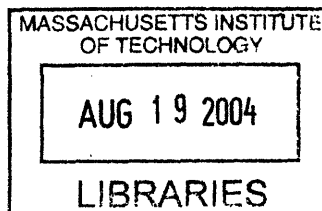


C. Forbes Dewey, Jr., Ph.D.
Professor of Mechanical Engineering and Bioengineering
Harvard-MIT Division of Health Sciences and Technology
Thesis Supervisor

Certified by:



Martha L. Gray, Ph.D.
Edward Hood Taplin Professor of Medical and Electrical Engineering
Co-Director, Harvard-MIT Division of Health Sciences and Technology



ARCHIVES



A New Class of Functions for Describing Logical Structures in Text

by

Ngon D. Dao

Submitted to the
Harvard-MIT Division of Health Sciences and Technology
on April 10, 2004
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Medical Engineering

ABSTRACT

Text documents generally contain two forms of structures, logical structures and physical structures. Loosely speaking, logical structures are sections of text that are both visually and semantically distinct. For example, a document may have an “introduction”, a “body”, and a “conclusion” as its logical structures. These structures are so named because each section has a distinct purpose in conveying the document’s logical arguments or intentions. Perfect machine recognition of logical structures in large collections of documents is an unsolved problem in computational linguistics.

This thesis presents evidence that a new family of functions on text segments carries information that is useful for differentiating document logical structures. For any given text segment, a function of this form is referred to as the *cadence*, and it is based on a new interpretation of the vector space representation that Gerard Salton introduced in 1975. Cadence also differs from the original Salton representation in that it relies on three heuristic transformations based on authorship, location, and term coherence.

To test the hypothesis that the cadence of a text segment carries information helpful to differentiating logical structures, a corpus was built containing 2800 documents with manually-annotated logical structures. Structures representing abstracts, introductions, bodies, and conclusions from this corpus were clustered with a k-means algorithm using cadence data. Precision and recall performances were computed for the results, and a chi-squared cross-tabulation test was used to determine the statistical significance of the clustering results. Precision and recall were highest for abstracts ($P = 0.931 \pm 0.025$, $R = 0.992 \pm 0.026$), followed by introductions ($P = 0.747 \pm 0.025$, $R = 0.802 \pm 0.026$) and conclusions ($P = 0.737 \pm 0.025$, $R = 0.813 \pm 0.026$), and lowest for bodies ($P = 0.876 \pm 0.03$, $R = 0.663 \pm 0.026$). These results suggest that cadence may have substantial promise for finding logical structures in un-annotated documents.

Thesis Supervisors: C. Forbes Dewey, Jr.,
Title: Professor of Mechanical Engineering and Bioengineering



ACKNOWLEDGEMENTS

My advisor – Prof. Dewey – has given me my most insightful years.

This research was made possible in part by the Defense Advanced Research Projects Agency and the Bioinformatics and Genome Training Fellowship.



Table of Contents

| | | |
|------------------|---|-----------|
| Section 1 | Introduction..... | 9 |
| Section 2 | Definition of Logical Structures and Related Works | 11 |
| 2.1 | Related Works on Physical Structures | 12 |
| 2.2 | Related Works on Logical Structures | 14 |
| 2.2.1 | Statistical Methods | 15 |
| 2.2.2 | Rule-based Methods | 16 |
| Section 3 | Definition of Cadence and Its Computation | 19 |
| 3.1 | A Hypothesis for Cadence..... | 19 |
| 3.2 | Definition of Cadence..... | 20 |
| 3.2.1 | Motivations for Heuristic Operators | 22 |
| 3.2.2 | The Coherence Operator | 23 |
| 3.2.3 | The Authorship Operator | 25 |
| 3.2.4 | The Location Operator | 25 |
| 3.3 | Remarks | 26 |
| 3.3.1 | Use of TF-IDF | 26 |
| 3.3.2 | Cadence and Stop Words | 26 |
| 3.3.3 | Error Due to Transformations..... | 26 |

| | | |
|------------------|---|-----------|
| 3.3.4 | Heuristic Operators are Not Commutative | 26 |
| 3.3.5 | Heuristic Framework | 26 |
| Section 4 | Empirical Evidence for Cadence | 27 |
| 4.1 | Test Corpus Description..... | 27 |
| 4.2 | Experimental Evidence for Cadence | 29 |
| 4.3 | Cadence and Logical Substructures..... | 33 |
| 4.4 | Relative Contributions to Cadence | 35 |
| 4.5 | Empirically Derived Aspects of Cadence..... | 37 |
| 4.5.1 | Rule of 5000 | 37 |
| 4.5.2 | The Topic Invariance of Cadence | 38 |
| 4.6 | Experimental Results with Entropy..... | 41 |
| Section 5 | Future Work..... | 45 |
| Section 6 | Concluding Remarks | 47 |
| Section 7 | Bibliography..... | 49 |

Section 1 Introduction

Text documents generally contain two forms of structures, logical and physical. Loosely speaking, logical structures are sections of text that are both visually and semantically distinct. For example, a document may have an “introduction”, a “body”, and a “conclusion” as its logical structures. These structures are so named because each section has a distinct purpose in conveying the document’s logical arguments or intentions. Physical structures, on the other hand, are conceptually different from logical structures. They are based solely on the physical layout of the document. Examples of physical structures are pages, columns, sentences, paragraphs, headers, etc. Physical structures by definition are demarcated with physical markers, whereas logical structures are demarcated by a combination of physical markers and conceptual boundaries.

The ability to discover logical structures automatically and perfectly within large collections of heterogeneous documents continues to be an unsolved problem in computational linguistics and computer science. The research described in this thesis tests the hypothesis that a particular characteristic of all text segments contains information that is useful in differentiating logical structures within documents. This thesis refers to this characteristic as *cadence*.

The ability to demarcate structure in text with a high degree of precision and recall would have many uses and would serve as a foundation for many text automation needs. For instance, one would be able to query a collection documents for all documents containing a particular phrase only if that phrase was in the “introduction” section. One would also be able to enhance document categorization.

This introduction is followed by 5 sections. Section 2 presents a formal definition for logical structures and reviews the literature on document logical structure analysis. Section 3 presents a formal definition for cadence and a method to compute it. Section 4 presents evidence that cadence contains information useful for differentiating logical structures. Section 5 offers remarks on future work and Section 6 concludes.

Section 2 Definition of Logical Structures and Related Works

The common notion that documents are made up of logical structures, or components that are topically cohesive, is intuitive. This intuition, however, has not led to a definition for logical structures that is both widely accepted and that does not rely on some subjective interpretation. This thesis uses the following working definition put forth by Summers:

The logical structure of a document consists of a hierarchy of segments of the document, each of which corresponds to a visually distinguished semantic component of the document [1].

Subjectivity arises when a high degree of discrimination is desired between one structure and another. Consider a large heterogeneous collection of text. The specification of all logical structures that appear in the collection will undoubtedly yield many overlapping structures. Some documents may consist of chapters which have titles. Each chapter may also be subsequently divided into sections and subsections. For other documents, however, the notion of chapters may not exist and these documents may be comprised of only sections, subsections and their titles. Chapters typically connote more significance than sections, and therefore can be thought of as distinct from sections. On the other hand, many documents exist where one can arguably replace chapters for sections or vice-versa and not change their logical intent.

Subjectivity also arises when seemingly similar logical structures are referred to by different names in different documents. For instance, the “Abstract” logical structure is referred to as a “Summary” for documents in the journal *Cell*. A knee-jerk reaction is that these two structures are identical except for their names, but the possibility exists that they are indeed different at some semantic and visual level. Without consulting every author of every structure, subjectivity is required to arbitrate, especially for large collections of text.

The notion of a set of logical structures for a set of documents is termed a document logical model. The number of possible logical structures for a large collection of documents make it unlikely that one document model can be defined that describes every document to a fine degree of granularity. To make the problem of discovering logical structures tractable, all of the methods reviewed in this section limit the scope of possible

logical structures in their respective document models. The work of this thesis also limits the scope of logical structures by using a document model consisting of only the abstract, introduction, body, and conclusion structures.

2.1 Related Works on Physical Structures

The research arena for automatic structure extraction for text dates back several decades. It has matured greatly with the confluence of mathematics, computer science, and more recently, computational linguistics. Electronic documents are generally thought of as having physical structures and logical structures. This thesis concerns logical structures, but a review of both areas is given here for completeness.

Previous work in discovering physical document structures falls mostly along rule-based or heuristic methods (of which syntactic and grammar-based are subsets) and statistical methods. Thorough surveys of many methods are given by Summers [1], S. Mao et al [2], R. Haralick [3], G. Nagy [4], and H. Fujisawa, Y. Nakano, and K. Kurino [5]. This section discusses the most important methods of physical structure analysis in more detail.

Rule based methods are typically top down or bottom up methods [6, 7, 8]. Wahl et al [9] performed some of the earliest work on page segmentation from scanned documents in 1982. They adapted the generic Run Length Smoothing algorithm found in other applications and applied to it text segmentation. Their algorithm was shown to be able to differentiate text, horizontal and vertical solid black lines, graphics and halftone images.

Fletcher and Kasturi [10] developed a bottom-up system in 1988 for use on OCR'ed (Optical Character Recognition) documents. The system uses connected component analysis that is not sensitive to font style and size. This was achieved by setting a threshold of eight connected pixels where the number eight was derived empirically. Each connected component is then circumscribed by rectangles and larger components are built on the inter-connection between eight smaller components. To separate text from graphics, a histogram of frequencies of components as a function of there areas is built. Manual inspection of the histogram is required to find a threshold that would separate text from graphics.

O'Gorman [11] reported a bottom-up system in 1993 called Docstrum that uses a nearest neighbor clustering technique to identify text lines and text blocks in OCR'ed documents. This system was a breakthrough in its time because it was nearly independent of skew-angle (page orientation) and line-spacing. Docstrum also did not assume a Manhattan layout*, which many of its predecessors required.

Another bottom-up method is the use of Voronoi diagrams as proposed by Kise et al in 1998 [12]. A Voronoi diagram of a collection of objects is a partition of the space around the objects into cells such that each point in the same cell of an object is closer to that object than any other objects in consideration. The algorithm worked by first creating a

* A layout style where blocks of text are separated by either horizontal or vertical demarcations.

highly granular Voronoi diagram of an OCR-ed document and then iteratively deleting edges of cells that satisfied specific area and distance requirements.

Another method proposed in 1998 was by A. K. Jain and B. Yu [13]. They proposed a method of segmenting pages by extracting segments that are connected to one another using a technique based on Block Adjacency Graphs. Although their method was reported to work well on business letters and forms, the emphasis of their algorithm was on scientific and technical articles where it can reliably differentiate text, tables, and images. They also proposed a document model tailored for technical documents. Their algorithm works by first converting OCR-ed documents into binary black and white images using a fixed threshold at 128 bits. The algorithm then breaks the image into a set of blocks by looking at each pixel row of the image and defining a new block if the previous pixel row's black run-length at that location is different than this row's current run-length. To identify text and non-text regions, the authors proposed the following heuristics:

A block is a text block if i) its height is less than 0.3 inch and the connected components in it are horizontally aligned and ii) its width is greater than 1 inch and the connected components in it have roughly the same height”.

A block is an image block if i) the smaller of its height and width is larger than 0.5 inches and ii) the ratio of the number of black pixels it contains to the its area in pixels is larger than 0.4.

A block is a table block if i) the top and bottom horizontal lines of the table have similar lengths (if those lines are present) and ii) the average height of its connect components is less than 0.3 inches.

Top-down methods for physical structures typically start with an entire document and iteratively split it into smaller units. The splitting of a unit stops when that unit satisfies some prior rule. Nagy & Seth [14] developed a top-down system called Gobbledoc and reported it in 1992. This system allows user to define strict layout rules for batches of similar documents. The authors give the following set of rules as an example: “The title lines are set in Melior 36/38 point boldface, centered...Title lines can be from 1 to four lines...The title line precedes the byline and the separation between them is 38-point leading.” In addition to these strict rules, Gobbledoc also allows the administrator to define grammar rules to label physical structures such as paragraphs and sentences. The combination of strict and grammar rules allow for an algorithm to guess where to demarcate any given document into X and Y segments. The final task of labeling each segment becomes trivial since each segment is defined by the rules it satisfies (i.e. if a segment satisfies all of the rules given above for title, it is a title).

Baird et al. [15] reported a top down system developed at AT&T Bell Laboratories in 1990. This system was designed to work specifically on Manhattan-type documents and uses a technique the authors called shape-directed covers. In general, top-down methods

have longer run times than bottom-up methods because they require recursive split-and-then-backtrack steps. Bottom-up methods do not incur these backtracking steps but pay a penalty on error rate because local evidence for decision-making is sparse at the initial phase of execution. With the system proposed by Baird, their technique tries to minimize the number of backtracking steps by identifying the most greedy split steps and executing them in order all the while maximizing the statistical support for their split criteria.

Rule-based approaches can become arbitrary unless they are constrained by an underlying grammar. Work by Chou in this area uses a combination of rules and statistical methods to introduce grammar by assuming that an underlying stochastic process for grammar is present. In the earlier work reported in 1994 [16], Kopec and Chou used a Markov model to generate likelihoods for possible outcomes for a set of predefined templates. The model used heuristically chosen transition probabilities and does not give suggestions for estimating these parameters. In his work reported in 2001 [17], Tokuyasu and Chou refined the notion of using communication theory with an algorithm named Turbo Recognition that was developed in collaboration with Microsoft Research. This algorithm used two Markov models to describe physical structures in the horizontal and the vertical dimensions. The authors implemented a prototype and demonstrated its effectiveness in decoding physical structures in the face of substantial OCR noise.

In 2003 [18], Kanungo and Mao reported on a system they developed which uses a segmentation algorithm that models a document's physical structure as a hierarchy. The segmentation algorithm is a weighted finite state automaton where the weights represent probabilities for each possible outcome. The probabilities are derived from ground truth data.

2.2 Related Works on Logical Structures

Discovering logical structures has generally been recognized to be more difficult than discovering physical structures. Overall performance of existing systems for logical structures continues to lag behind those for physical structures [19]. Many people have proposed many solutions to varying aspects of this problem. In 1975, G. Salton proposed one of the first mathematically rigorous methods for computing the similarity between text segments in a seminal paper titled "A Vector Space Model for Automatic Indexing [20]. We give a thorough review of this work and of Salton's subsequent work in the 1970's because it is the foundation for the concept of cadence.

Salton first developed his vector space model for text with the goal of developing decision rules for determining which terms in documents should be indexed. The intuition is that not all terms should be indexed equally because not all terms contributed the same to the precision-recall performance of an information retrieval (IR) system. Two decades before, Luhn at IBM demonstrated that scoring terms in proportion their frequencies in documents can increase retrieval performance over systems where all terms were treated equally [21]. This is particularly true when there are moderate numbers of these high frequency terms and when they are not uniformly distributed across the document collection. For certain document collections where these two conditions are not met, the retrieval performance is actually worse. This observation led Karen Sparck Jones of

Cambridge University to propose an alternative weighting method called Inverse Document Frequency which emphasized terms which appears in few documents [22]. The notion here is that these terms are specific to a small and exclusive set of documents, and so they are better positioned to distinguish these documents from the rest of the collection. It turns out that performance gains afforded by this method are also collection dependent. Salton's concept of discrimination value analysis was a breakthrough in its time because it was the first method of indexing that was not collection dependent. His technique is also computationally simple. Salton first describes the concept of discriminating value analysis in his 1971 paper titled "Experiments in automatic thesaurus construction in information retrieval" [23] and in another paper in 1975 titled "A theory of term importance in automatic text analysis" [24]. These two papers demonstrated significant precision gains for an IR system when it optimizes the discrimination properties of terms.

The use of the vector space model or other methods for logical structure identification falls along two lines [1, 2]: statistical methods or rule-based methods. Rule-based methods can be further divided into heuristics or semantic and grammatical methods.

2.2.1 Statistical Methods

Tateisi and Itoh [25] developed a system in 1994 that uses stochastic syntactic analysis to classify components of a document as text or graphic. If a component is text, it is further classified as into several possible logical structures, such as headings, first and last lines of a paragraph, continuation lines of list items, ordinary lines, etc. The syntactic nature of this method first classifies a component into classes where each class is defined by a set of rules (or grammar). The grammar is supposed to be similar to that used to classify terms into parts of speech, hence the name syntactic. Associated with each rule is the probability of that rule being associated with a particular line. The authors do not present details on how the probabilities are chosen.

In 1997, Brugger, Zramdini, and Ingold [26] proposed a document model for logical structures based on the concept of n-grams. Their method relies on the specification of a tree where each node represents an instance of logical structure. The tree requires that $n-1$ seed nodes are constructed and these seed nodes determine the probabilities of the n^{th} node. To derive the probabilities for each possible outcome of a structure, the authors use a generalized means function.

Dengel and Dubiel [27] describe a system for identifying the logical structures common in business letters in 1996. The system requires a training-set of documents as inputs and clusters their components into a concept hierarchy. The concept hierarchy is defined *a priori* and contains for each concept (logical structure) a set of attributes and their values that most differentiates one concept another in the hierarchy. Document components are classified into this hierarchy by examining how many attributes overlap that of a concept.

Ahonen [28] implemented in 1996 the first known application of grammatical inference to the problem of inferring document type definitions for SGML documents. The method infers knowledge about element order, whether elements are optional or required, and

whether elements were iterative. Another approach is given by Young-Lai also in 1996 [29]. The difference between the two approaches is that the latter assumes a stochastic model for errors – such as misspellings – present in large collections of documents. The work by Ahonen, however, assumes that no such exceptions are present.

Key and Wong [30] introduced another statistical inference method in 2001 building on methods proposed by Ahonen and Young-Lai. The method proposed by Key and Wong uses a probabilistic finite state automaton. They derive the parameters for these transition probabilities empirically via a test collection and its transition frequencies.

2.2.2 Rule-based Methods

The literature contains many descriptions of rule-based methods. This section only describes some of the more important ones.

Bayer and Walischewski [31] describe a system developed at Daimler-Benz Research in 1995 which uses a semantic network to extract logical structures from business letters. The authors developed the system using their own semantic network to specify relationship for components in their document model. They also extended the notion of the traditional semantic network by attaching to each structure grammatical attributes that specifies spatial constraints relative to other structures.

In 1997, Wenzel [32] developed a semantic-based logical structure extraction system for use with business letters. The core component of this system is a pattern matcher that tries to detect key phrases to match against pre-defined set of logical structures. The system allows users to declare these logical structures and phrases via a pattern language developed by the author.

In 1993, Saitoh, Tachikawa, and Yamaai [33] reported their system for detecting physical components and classifying these into logical structures. The system classifies objects as captions, headers, footers, or bodies. Objects are classified on the basis of satisfying simple rules. An example rule is that a text component is a caption if it is close to a diagram or is it near the bottom of diagram-frames or tables. The authors do not describe how they set thresholds for each rule; e.g., what does it mean to be close to a diagram?

Niyogi and Srihari [34] developed a system in 1995 that segments documents into physical structures, groups these physical structures into logical units, and assigns reading orders for the logical units. The system uses hierarchy of rules to achieve organizing rules into manageable classes. The authors reported that their system uses 160 rules. 114 of these rules are designed specifically for newspapers and the remainder can be used with other document types.

Salton et al. [35] proposed a text decomposition method in 1996 based on his vector space representation model. The method proposed a unified way to discover functionally homogeneous excerpts known as segments and groups of not-necessarily contiguous excerpts known as themes. A segment is defined as a contiguous piece of text that is internally linked but is largely disconnected from adjacent text. A theme is a group of text

excerpts that address a common topic. The method computes similarity scores between text excerpts. To identify segments, groups of 5 neighboring paragraphs are examined one at a time for similarity above a defined threshold of 0.2. A segment is found if two or more of the paragraphs in a group are related by a similarity greater than the threshold. To find text themes, a relationship map is constructed that represents excerpts as nodes and these nodes are connected by links if they share a similarity value greater than the threshold value of 0.2. All possible triangles in this map are then considered by computing centroid vectors as the averages of the three vectors in the respective centroids. Similarities of centroids are then computed and centroids are merged if their similarities exceed a given threshold. When no further merging is possible, themes are identified as the final set of triangles remaining.

Lin, Niwa, and Narita [36] developed a clever system in 1997 to extract the logical structure of books by taking advantage of information in their table of contents. Their system automatically locates the table of contents, extracts heading and pagination information, and then uses this information to perform text matching within the body of the book.

Kochi and Saitoh [37] reported the development of a system in 1999 where document templates are matched against individual documents to extract logical structures. The system assumes that all documents to be processed come from a known template. The matching algorithm computes a distance measurement between each structure and template features and matches documents to templates where the global distance for all features is minimum. The authors do not describe their method for specifying weights. A similar algorithm is described by Summers in 1998 [1]. The work by Summers is more general in the sense that it deals with a hierarchy of logical structures as well as many more templates. Summers also describes but does not implement ways in which this algorithm can be modified to use machine learning techniques to achieve higher performance.

In 1999, Ishitani [38] from Toshiba Corporation developed a system for detecting logical structures by dividing the process into four modules. Each module relies on simple heuristics. Each of the four modules can collaborate with other modules to extract logical structures where there are imperfect layouts and/or errors. The first module classifies each text line into one of five categories: normal, indented, centered, new line, or previous to new. The second module labels connected text line as one of four structures: title, paragraph, list, or formula. The third module segments connected lines that could not be classified into any of the four structures into smaller segments for classification by the second module. The fourth module groups objects that have been over-segmented. An example of collaboration between modules is when a text component is not classifiable into a category by module one, so this module sends the component to either module 3 or 4 for modification before trying again.

Section 3 Definition of Cadence and Its Computation

There are many ways to describe a segment of text. One could describe it at a semantic level where one speaks of its topic or intention. The alternative is to describe it statistically, such as how many terms it has, the frequency of each term it contains, or as in the case of this thesis, its cadence. This section introduces cadence as a new way of describing text segments. The next section tests the hypothesis that cadence can be useful in differentiating text segments from differing logical structures.

3.1 A Hypothesis for Cadence

Informally, the cadence of a text segment is the ‘importance’ of its words as a function of fractional position in the segment. One can think of cadence as a plot of the importance of each term in the segment versus its relative position; the importance of the first term would be plotted at position 0% followed by that of the second term, that of the third term and so forth until the last term’s importance is plotted at position 100%. We are assuming for now that each term’s importance can be specified somehow. Consider a hypothetical text segment that has important terms in the beginning, not-so-important terms in the middle and it ends with important terms. Its cadence would be shaped in the form of the U similar to the cartoon of Figure 1. One can imagine that other segments can have more complex or less complex cadences. Some may be shaped in the form of sinusoids. Some may be flat, and some may be complex combinations of different profiles.

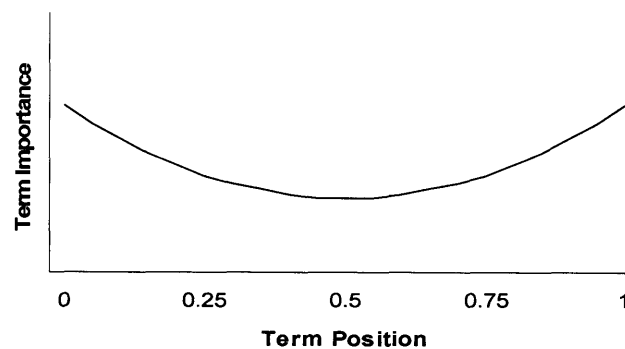


Figure 1: Illustration showing a hypothetical cadence for a text segment where the beginning and ending terms are more important than the middle terms.

The notion of cadence as presented thus far is fairly simple. If we assume that all terms in a language can be assigned some importance, then it is clear that there exists a cadence for any text segment from that language. Whether or not that segment is meaningful does not matter. The only requirement is that the segment contains one or more terms and that each term has a definable importance*. Cadence may be intuitive to grasp, but it is not obvious why cadence contains useful information. If this is not obvious, one may ask “where did the motivations for representing text in this way come from?” It turns out that this thesis is a result of fortuitous observations that under certain circumstances, text segments that have the same logical structures show similar cadence plots. The reason for this is not clear, and this thesis does not attempt to answer that question. Instead, this thesis hypothesizes that cadence can be useful, and it presents statistically significant evidence to validate this hypothesis.

3.2 Definition of Cadence

Consider a set of text documents where terms have been stemmed, stop words† have been removed, and the authorship of each document is known. For these documents, one can arbitrarily extract n text segments where each segment is a contiguous subtext. For these segments, one can define vectors of term importance of the form

$$\vec{v}_i = (w_{i1}, w_{i2}, \dots, w_{im_i}) \quad (1)$$

where the i indicates the i^{th} segment extracted from this corpus and the w 's are term importance weights assigned to each term in the segment. The order of each weight – as indicated by the second subscript – corresponds to the same order that its respective term appears in the text. Note that this formulation is sufficiently general to accommodate differing methods of computing term importance. Note also that these segments may contain different number of terms and therefore may occupy spaces of different dimensions; the dimension of a segment will vary according the number of terms in that segment and so the dimension is designated as m_i . To compare vectors, one must bring them into the same dimension. One method to do this is to transform each vector \vec{v}_i into a continue function $f_i(x)$ from the interval $[0,1]$ to the reals. $f_i(x)$ is a function which interpolates the values in \vec{v}_i . Formally,

$$\text{Let } y = x(m_i - 1) + 1$$

$$f_i(x) = \vec{v}_i \llbracket y \rrbracket + (y - \lfloor y \rfloor)(\vec{v}_i \lceil y \rceil - \vec{v}_i \llbracket y \rrbracket) \quad (2)$$

* It is theoretically possible to have a text segment with one or more terms, but the cadence is undefined. These cases are possible because the formal definition of cadence requires the removal of stop words from a segment. Text segments consisting of only stop words will therefore have no definable importance.

† See 3.3.2 for more details on stop words.

The notations $\lfloor y \rfloor$ and $\lceil y \rceil$ indicate the lower and upper integer bounds, respectively, of the real value $x(m_i - 1) + 1$. $\bar{v}_i[\lfloor y \rfloor]$ and $\bar{v}_i[\lceil y \rceil]$ uses array notation to indicate the elements of the vector \bar{v}_i at the lower and upper integer bounds.

Figure 2 illustrates an example of how the function $f_i(x)$ and the vector \bar{v}_i are related for a case where the dimension of \bar{v}_i is ten:

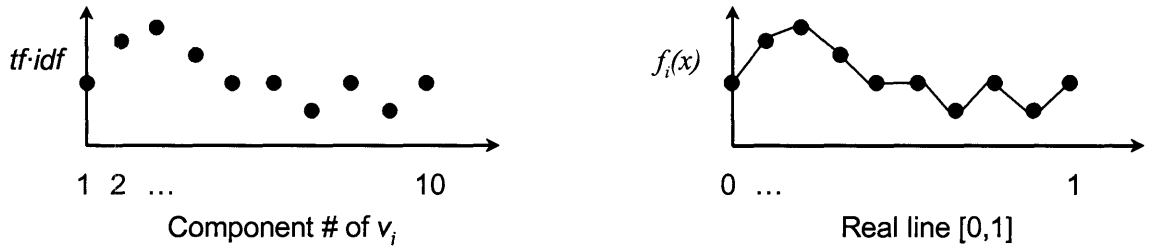


Figure 2: Illustration showing that $f_i(x)$ is the interpolation of the tf-idf weights over the real line from 0 to 1. This illustration uses the tf-idf term weighting scheme.

The functions $f_i(x)$ can then be operated on by three heuristic operators $H_{ij}(\)$ which are specific to each segment as follows:

$$F_i(x) = H_{i1}(H_{i2}(H_{i3}(f_i(x)))) \quad (3)$$

These operators are described in more detail the next subsection. For now, assume that these operators together produce a continuous function $F_i(x)$. Defining this new function $F_i(x)$ for the set of n segments would result in the following set of functions

$$\begin{aligned} F_1(x) &= H_{11}(H_{12}(H_{13}(f_1(x)))) \\ &\quad \vdots \\ F_n(x) &= H_{n1}(H_{n2}(H_{n3}(f_n(x)))) \end{aligned}$$

A function $F(x)$ of this form is defined as the *cadence* of a text segment. For any given segment, one can define a class of these functions by varying the method one chooses to weight term importance. In much of the experimentation for this research, the *tf-idf* weighting scheme was used. For the rest of this thesis, we use the notation $\bar{v}_i = (tf \cdot idf_{i1}, tf \cdot idf_{i2}, \dots, tf \cdot idf_{im_i})$ unless otherwise noted. Although most of the experiments for this research uses the *tf-idf* metric, this thesis also presents preliminary evidence that other metrics, specifically one based on term entropy, may be equally effective at differentiating logical structures.

The only items in the definition of cadence that require further specification are the heuristic operators $H_{ij}(\)$.

3.2.1 Motivations for Heuristic Operators

The motivation for heuristic operators came from fortuitous observations of results from exploratory experiments. Results from those experiments formed two plots of raw *tf-idf* weights unaltered by heuristics. These plots are shown in Figure 3. The plots show that very slight similarities exist between plots sharing certain characteristics. Specifically, it

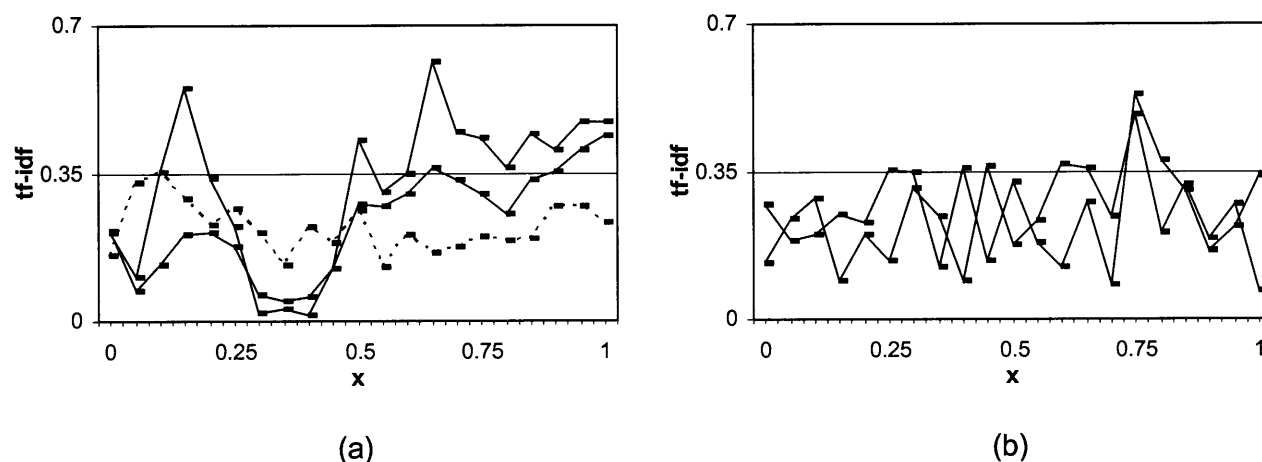


Figure 3: Plot (a) and (b) are plots of *tf-idf* weights for text segments as a function of position in their segments. Plot (a) is for three abstracts from two authors. Note that the two solid lines from the same author look slightly different than the dashed line from a different author. Plot (b) shows the weights for two text segments that are from the introductions of two different documents sharing the same author. Finally, note how the plots in (a) differ from those in (b) and that those in (a) come from a different region of documents than those in (b).

was noted that if two abstracts came from the same author, their *tf-idf* profiles were sometimes more visibly similar than for two segments from different authors. Similarly, if two segments appear roughly within the same region of their respective documents (beginning, middle, end), it was noted that their profiles sometimes shared more similarities than if they had come from different regions. At the time, these observations were not statistically significant nor did they infer causality between authorship (or location) and profile characteristics with reasonable confidence. They did, however, bring up the possibility of applying a series of heuristics to amplify hidden information present in *tf-idf* profiles. A major section of this thesis will show that these early observations have since been validated with statistically significant findings.

The third heuristic operator was not inspired by any experimental findings. It simply has to do with a hunch that segments representing the same logical structures share the same level of terminological coherence. For example, one may expect abstracts to be generally more terminologically cohesive than bodies. One may also expect introductions and

conclusions to be more cohesive than bodies, but less so than abstracts. The motivation for this heuristic is that if these notions are true, taking advantage of them may help differentiate abstracts from introductions and introductions and conclusions from bodies.

3.2.2 The Coherence Operator

The operator $H_{i1}(\)$ is the first operator for the i^{th} segment. It is called the coherence operator and its goal is to introduce terminological coherence information into the tf-idf vector.

One can think of this operator intuitively by imagining two text segments representing the same logical structure from different documents. Imagine also that these segments address completely different fields of interest. An example would be an abstract from an article on physics and an abstract from an article on anthropology. Imagine examining the nouns in each segment and counting the ones that are hyponyms or hypernyms* of other nouns in the same segment. In a sense, this count is a measure of the terminological coherence of a text segment. The higher count, the more coherent the segment is because the same concepts are being referred to repeatedly by synonyms. Likewise, if the count is low, most nouns are referring to completely different concepts and none can be said to be a synonym of another.

For our thought experiment, what if it was true that the ratio of the number of coherent nouns to the number of terms in a segment was about the same for our physics and anthropological abstracts? More generally, what if it was true that similar logical structures tend to share this ratio and that the raw tf-idf weights do not sufficiently draw out this fact? For example, it is not far-fetched to imagine abstracts to be generally more terminological cohesive than bodies. To “inject” this information into the functions $f_i(x)$ to aid in the discrimination of different logical structures is precisely what the coherence operator is designed to do.

The operator takes as its arguments the functions $f_i(x)$. Recall that these functions are simply the raw tf-idf vectors transformed onto the real line. The operator is applied using the following algorithm:

- 1) Define the raw vectors

$$\bar{v}_i = (tf \cdot idf_{i1}, tf \cdot idf_{i2}, \dots, tf \cdot idf_{im_i}) \quad (4)$$

for a corpus of documents where terms are stemmed and stop words have been removed. If there are n segments from this corpus, there would be n of these vectors. For each vector \bar{v}_i , define $f_i(x)$ as in Equation (2).

* According to the WordNet glossary: “A hypernym is a term used to designate a whole of a class. Y is a hypernym of X if X is a kind of Y. A hyponym is a term used to designate a member of a class. X is a hyponym of Y if it is a kind of Y”.

2) For each function $f_i(x)$, compute a *coherence ratio* $r_i = c_i/m_i$. Recall that m_i is the number of terms in the i^{th} segment. c_i is the number of synonyms in the i^{th} segment. c_i can be found with the following substeps:

- a) For each noun* term in the segment, query the WordNet 2.0 database for its hypernyms and hyponyms.
- b) Collect into a set S_i all unique terms as a result of the WordNet queries using all of the nouns in the i^{th} segment.
- c) c_i is the number of terms in the segment that also appear in S_i .

3) Histogram all functions $f_i(x)$ into 5% bins according their coherence ratio. The first bin is for functions where $0 \leq r_i < 0.05$, the second group for functions where $0.05 \leq r_i < 0.10$ and so forth until $0.95 \leq r_i \leq 1$. The bins quantized to 5% is arbitrary. No studies have been done to study the effects of using a different binning strategies.

4) For each bin, find the mean function of that bin

$$\bar{f}(x) = \frac{1}{g} \sum_g f_i(x)$$

where g is the number of function in the respective bin.

5) Define the following difference function for each function $f_i(x)$, considered with respect to its bin mean, $\bar{f}(x)$:

$$\bar{f}c_i(x) = f_i(x) - \bar{f}(x)$$

6) Finally, the $H_{il}(\)$ operator for all functions $f_i(x)$ is defined as

$$H_{il}(f_i(x)) = f_i(x) - T_c \bar{f}c_i(x) \quad (5)$$

* To identify a term as a noun, one could use a parts-of-speech tagging algorithm. This thesis does not use such a system. Instead, each term is fed to the WordNet 2.0 database, and a term is identified as a noun if WordNet identifies one of its senses as a noun. This method over-recognizes nouns, but this is acceptable since erring in this way is in line with the goal of injecting this type of information into the raw vectors. A disadvantage of this approach is that it may not recognize domain specific nouns, such as the names of proteins.

where T_c is called the *coherence attractor*. It is a scalar on the interval $[0,1]$ and chosen to be the same for all text segments. This thesis arbitrarily chooses the value to be 0.10.

Intuitively, this algorithm preserves the mean and lowers the variance of each bin. That is, it shifts each points in the function $f_i(x)$ closer toward the corresponding point of its bin mean function $\bar{f}(x)$.

3.2.3 The Authorship Operator

The operator $H_{i2}(\)$ is the second operator for the i^{th} segment. It is called the authorship operator. Its goal is to incorporate author information into the image of the $f_i(x)$ function under the coherence operator. The operator is applied using nearly the same algorithm used for the coherence operator. The inputs to this operator are the functions $H_{i1}(f_i(x))$:

1) Bin all functions $H_{i1}(f_i(x))$ according to their authorship. In the case of multi-author documents, pick a convention for which authors to use (first, second, etc.). The algorithm implemented for this research used only the first author*.

(2) The remaining steps are similar to steps (5) and (6) for the coherence operator. That is, find the mean function for each bin and call it $\bar{f}(x)$. Then find the difference function $\bar{f}a_i(x)$ for each $H_{i1}(f_i(x))$ relative to its bin mean. Finally, the location operator is defined as

$$H_{i2}(H_{i1}(f_i(x))) = H_{i1}(f_i(x)) - T_a \bar{f}a_i(x) \quad (6)$$

where T_a is called the *authorship attractor*. This thesis chooses a value of 0.10 for T_a .

3.2.4 The Location Operator

The third operator $H_{i3}(\)$ is called the location operator. It introduces information about where in the document the i^{th} segment came from, relative to the document beginning. This thesis uses segmentation quantized to 5%. Perhaps finer or coarser segmentations are more informative. The author is not aware of a scheme for optimizing segmentation in relation to cadence.

Some text segments will inevitably straddle bin demarcations. One may use an arbitration method to bin these straddlers to one side or the other. This thesis counts number of terms on each side of the demarcation and bins segments to the side with the most terms. The steps for the location operator are the same as the authorship operator. This operator bins all functions $H_{i2}(H_{i1}(f_i(x)))$ according to whether or not their corresponding text

* The choice of only using the first author leaves open the question as to whether an optimum choice exists. Perhaps using more than one author or using only the last author may yield better results.

segments came from the first 5%, second 5%, third 5%, or so forth of their documents. The operator is defined as

$$H_{i3}(H_{i2}(H_{i1}(f_i(x)))) = H_{i2}(H_{i1}(f_i(x))) - T_i \overline{f}_i(x) \quad (7)$$

where T_i is the *location attractor* and $\overline{f}_i(x)$ is the difference function defined analogously to the ones for coherence and authorship. This thesis chooses the value 0.10 for T_i .

3.3 Remarks

There are several remarks about the prescribed formulation of cadence and the use of heuristics.

3.3.1 Use of TF-IDF

As noted previously, much of the research in this thesis uses the tf-idf weighting scheme for term importance. In theory and in practice, one could use another term scoring scheme. The question remains unanswered as to what is the optimal scheme for differentiating logical structures. Section 4 below first presents evidence for the utility of cadence using tf-idf and then follows it with preliminary data using another metric based on term entropy.

3.3.2 Cadence and Stop Words

Throughout section 2.0, stop words are mentioned as being removed from segments before the construction of \vec{v}_i . This thesis uses the stop word list from the SMART retrieval system developed by Salton in 1971 [39]. It should be noted that the SMART stop word list can remove more than it should.

3.3.3 Error Due to Transformations

The transformation of the \vec{v}_i vector into a real-valued function $f_i(x)$ assumes that it is meaningful to make something continuous out of something inherently discrete. There is no deep reason why this should be done at all. One should recognize, however, that in doing this transformation, one is “straying” from the data. One could theoretically minimize this type of error by comparing cadences near points where data actually exists. A thorough treatment of this type of error would be of interest in the future.

3.3.4 Heuristic Operators are Not Commutative

The authorship and location operators are not algebraically commutative. That is, $H_{i3}(H_{i2}(H_{i1}(\vec{v}_i))) \neq H_{i2}(H_{i3}(H_{i1}(\vec{v}_i)))$. The author has not experimented with using a different order and is not aware of a reason to prefer one order over the other.

3.3.5 Heuristic Framework

Does there exist an over-arching framework for deriving or choosing good heuristic operators? The author believes that this continues to be an unsolved question. The author also believes that an understanding of this topic will be of fundamental importance to the advancement of cadence in the future.

Section 4 Empirical Evidence for Cadence

One hypothesis of this thesis is that cadence carries information useful for differentiating segments representing different logical structures. Formally, the two hypotheses tested are:

H_0 : Cadence carries no useful information in differentiating abstracts, introductions, bodies, and conclusions.

H_1 : Cadence carries information useful in differentiating abstracts, introductions, bodies, and conclusions.

This section presents data from testing these hypotheses. The general structure of the test is as follows:

- 1) A corpus of about 2,800 documents was assembled and manually annotated. The annotations demarcate the abstract, introduction, body and conclusion of each document. Not all documents contain all structures.
- 2) Equal numbers of text segments from each of the four logical structures were randomly selected from the corpus.
- 3) A k-means clustering algorithm with $k=4$ was used to discover 4 natural groups of segments based on cadence information.
- 4) A Chi-squared test is used to assess the significance the clustering results.

This section first describes the corpus and then presents data from the hypothesis test. This section also presents experimental data comparing various aspects of cadence.

4.1 Test Corpus Description

The annotated corpus used for the hypotheses test in this section contains over 2,800 documents. The documents came from two main sources. The first source is from the

standard Cranfield, Medlars, and Times collections of documents [40, 41]. The Cranfield collection contains 424 abstracts in aerodynamics. The Medlars collection contains 450 abstracts in medicine. The term Medlars is now used to denote a file format for documents in the PubMed database. The Medlars collection referred to in this thesis is the 1969 collection. The Times collection contains 425 world news articles from the 1963 editions of *Time Magazine*. These three corpora have been used as standard research corpuses since the 1970's.

The second source was donated from EBSCO Industries, Inc. This collection contains 2,350 documents spanning a large range of topics, from information technology technical support to lawn mower repair instructions. Each document from this collection contains an abstract, introduction, body, and conclusion. Table 1 summarizes the contribution of each source to the annotated test corpus.

| Source | Abstracts | Introductions | Bodies | Conclusions |
|--------------|--------------|---------------|--------------|--------------|
| Cranfield | 424 | - | - | - |
| Medlars | 450 | - | - | - |
| Time | - | 450 | 450 | 450 |
| EBSCO | 2,350 | 2,350 | 2,350 | 2,350 |
| Total | 3,224 | 2,800 | 2,800 | 2,800 |

Table 1: Contributions from the sources of the annotated test corpus.

Figure 4 shows that the corpus obeys the usual form of Zipf's Law in that word frequencies obey a power law distribution with exponent approximately -1. The existence of this law indicates that words in this corpus are distributed similarly to generally accepted corpora in English.

The annotated test corpus is actually a subset of a larger un-annotated corpus of 48,000 documents. The *tfidf* weights of each non-stop term in the annotated documents are computed relative to the entire 48,000 documents. That is, the *tfidf* weight of the i^{th} term in the j^{th} document is

$$tf_{ij} \cdot idf_{ij} = n_{ij} \cdot \log \left(\frac{N}{D_j} \right) \quad (8)$$

where n_{ij} is the number of occurrences of the i^{th} term in the j^{th} document, N is the corpus size (in this case equal to 48,000), and D_j is the number of documents in the corpus in which the i^{th} term appears at least once. In theory, one can compute term weights relative

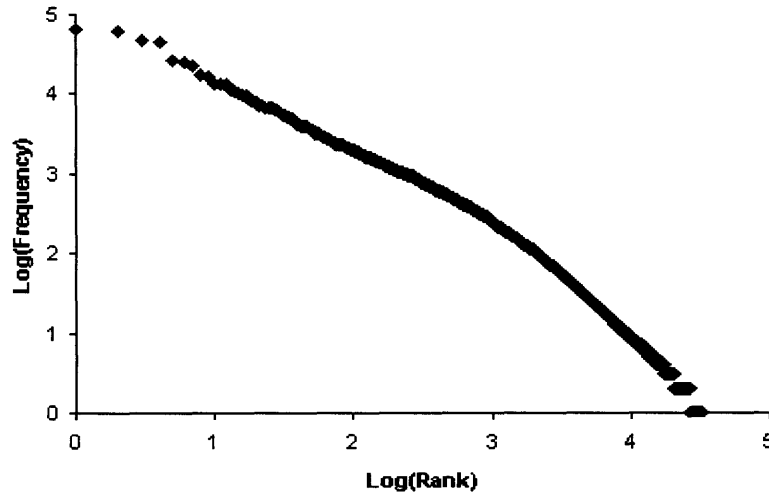


Figure 4: Zipf plot of for terms in test corpus.

to just the annotated corpus, but it turns out that for a corpus size under 5000, the utility of cadence is substantially diminished. It is unclear how or why this phenomenon exists. Experimental results demonstrating it is presented in 4.5.1

4.2 Experimental Evidence for Cadence

1000 abstracts, introductions, bodies and conclusions were randomly selected from the annotated corpus for a total of 4000 segments. Figure 5 shows the cadence plots for a subset of these segments*.

The 4000 cadences were randomly assigned to four unlabeled clusters. A k-means clustering algorithm with $k=4$ was applied to the clusters. Results are shown in Table 2 along with precision and recall statistics.

Cluster 1 contains all cadences that are classified by the k-means algorithm as abstracts. Cluster 2 contains cadences classified as introductions, cluster 3 for bodies, and cluster 4 for conclusions. The table highlights cadences that were correctly classified in red.

* Microsoft Excel limits the number of series on one graph to 255 so these graphs show cadences for 255 randomly selected segments.

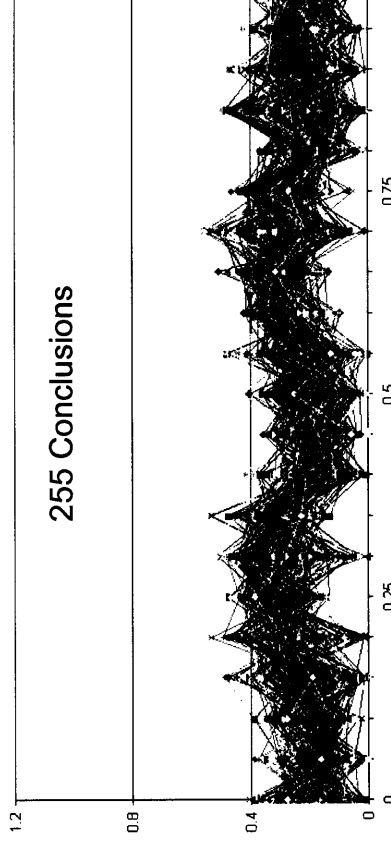
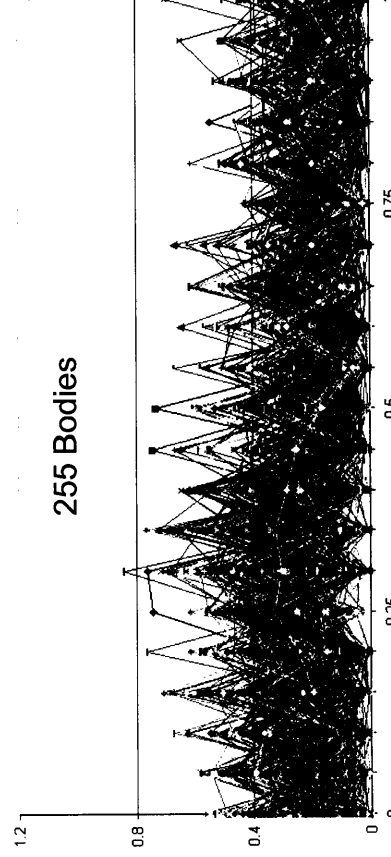
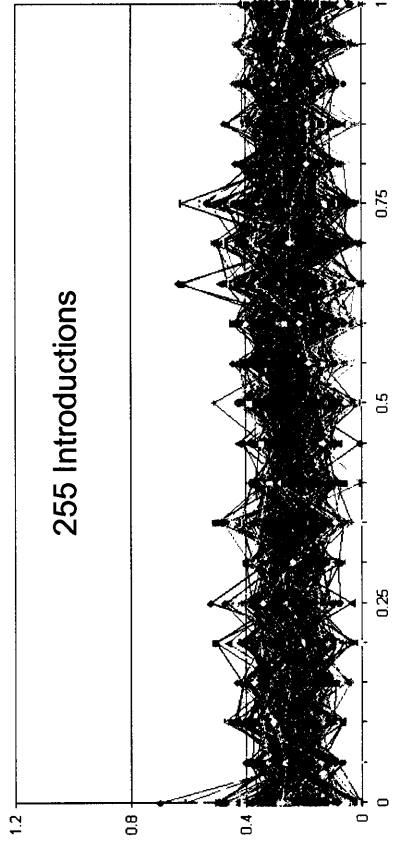
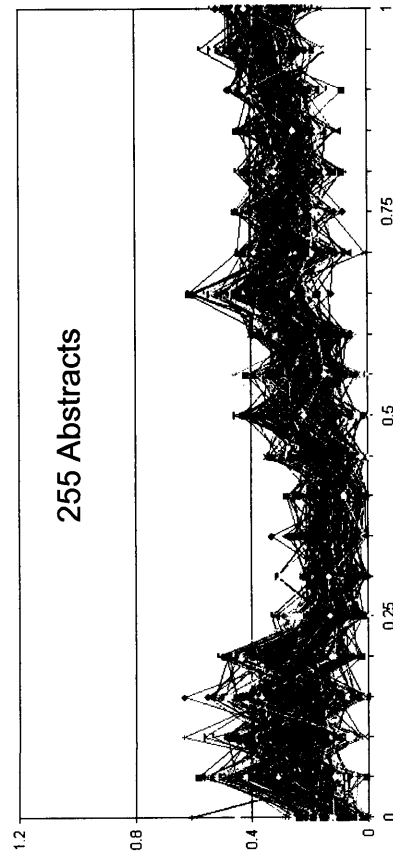
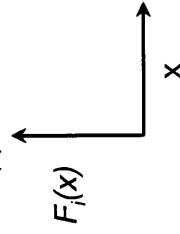


Figure 5: Cadence plots for 255 segments from each of the four structures being tested. Plots are limited to 255 cadences per plot because this is the limit of Microsoft Excel. The axes of the plots are $F_i(x)$ and x where $F_i(x)$ is the cadence at x and x is the relative position in segment i :



| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|---|----------------------|----------------------|----------------------|----------------------|
| 1 | 995 | 39 | 23 | 10 |
| 2 | 3 | 755 | 163 | 100 |
| 3 | 0 | 48 | 680 | 31 |
| 4 | 2 | 158 | 134 | 859 |
| Precision | 0.933 ± 0.025 | 0.739 ± 0.025 | 0.896 ± 0.025 | 0.745 ± 0.025 |
| Recall | 0.995 ± 0.026 | 0.755 ± 0.026 | 0.680 ± 0.026 | 0.859 ± 0.026 |
| Balanced-F | 0.963 | 0.747 | 0.773 | 0.798 |
| Global Precision: 0.822 ± 0.013 | | | | |
| Chi-squared Statistic: 7381 | | | | |
| Chi-squared p-value (15 degrees of freedom): less than 0.005 | | | | |

Table 2: k-means clustering results with parameter k=4.

Results shown in Table 2 is used to test between the hypotheses:

H_0 : Cadence carries no useful information in differentiating abstracts, introductions, bodies, and conclusions

H_1 : Cadence carries information useful in differentiating abstracts, introductions, bodies, and conclusions.

A chi-squared statistic T for these results is determined using the formula

$$T = \sum_{i=1}^{i=4} \sum_{j=1}^{j=4} \frac{(N_{i,j} - 250)^2}{250} = 7381$$

where $N_{i,j}$ is the number of segments in the (i,j) entry of the table. The value 250 is the expected number of segments in each cluster if each segment had the same likelihood of being assigned to any of the four clusters. The corresponding p-value with 15 degrees of freedom is significantly less than 0.005. This suggests that the clustering results in Table 2 have less than 0.005 probability of occurring under the scenario of uniformly random group assignments. Since the performance metrics are very good, one can infer H_1 and reject H_0 .

Precision and recall confidence intervals are also reported in the table. Precision for the j^{th} column is computed as:

$$precision_j = \frac{\max_i(n_{i,j})}{\sum_i n_{i,j}}$$

where j is the column index and it represents the segment type (abstract, introduction, body, conclusion), i is the row index and it represents cluster number, and $n_{i,j}$ is thus the number of segments from the j^{th} type that was clustered into the i^{th} cluster. $\max_j(n_{i,j})$ is the largest value over all columns in the i^{th} row.

Recall for the j^{th} column is computed as:

$$recall_j = \frac{\max_i(n_{i,j})}{1000}$$

where $\max_i(n_{i,j})$ is the largest value over all rows in the j^{th} column. This numerator is divided by the constant 1000 because the experimental setup chooses 1000 of each type of segments.

For each of the four types of segments, a Balanced-F score is computed as

$$BF_j = \frac{2 \cdot precision_j \cdot recall_j}{(precision_j + recall_j)}$$

Confidence intervals for precision and recall are computed as follows:

- 1) Treat the cadences of a logical structure type (abstract, introduction, etc.) as independent Bernoulli random variable X_i 's, where $i = 1 \dots 1000$.

$$\Pr(X_i = 1) = r$$

$$\Pr(X_i = 0) = 1 - r$$

$X_i = 1$ when the i^{th} cadence is classified correctly and 0 otherwise. r is the unknown true recall performance of the clustering algorithm based on cadence.

- 2) Define $recall = \frac{1}{1000} \sum_{i=1}^{1000} X_i$. By the Central Limit Theorem, $recall$ can be approximated as a Normal random variable with mean $\mu = r$ and variance $\sigma^2 = \frac{r - r^2}{1000}$. The variance is a function of the unknown r . One can bound the variance by noting that the maximum of the function $r - r^2$ is $\frac{1}{4}$ at $r = 0.5$. Thus we use the maximum variance of $\sigma^2 = \frac{1}{4 \cdot 1000}$.

- 3) Convert $recall$ into a Standard Normal random variable by subtracting its mean and dividing by its standard deviation.

$\frac{recall - r}{\sigma} = \sqrt{4 \cdot 1000}(recall - r)$ is distributed as a Standard Normal random variable. The confidence interval for r at 90% confidence is thus

$$\left[\frac{-1.645}{\sqrt{4 \cdot 1000}}, \frac{+1.645}{\sqrt{4 \cdot 1000}} \right] = [-0.026, +0.026]$$

The confidence interval for precision can be approximated using the same procedure where the constant 1000 is replaced by the total number of cadences in each cluster.

4.3 Cadence and Logical Substructures

The previous section presented evidence that supports the notion that cadence can differentiate text segments representing different logical structures. What if the same k-means algorithm is applied to a set of segments from only one logical structure? That is, can cadence differentiate abstracts from abstracts, introductions from introductions, bodies from bodies, and conclusions from conclusions?

Table 3 through Table 6 show the results of four different experiments (one for abstract, one for introductions, etc.). In each experiment, 1000 segments of the same type of logical structure were randomly selected from the corpus and randomly assigned to four unlabeled clusters. Then the clustering algorithm with parameter $k=4$ was applied to the clusters to produce four output clusters. The value of $k=4$ was not chosen for any particular reason other than it is the same value used in experiments presented previously.

The experiment for abstracts indicates that cadence cannot differentiate one abstract from another in a statistically significant manner. A reasonable conclusion one can make is that cadence cannot differentiate substructures within abstracts. Another possible conclusion could be that cadence is not finding different types of abstracts. Similar conclusions can be made for introductions and conclusions. For bodies, the assignments are not consistent with a non-informative random process because the p-value is too low. There is a possibility that cadence is detecting meaningful substructures within body segments or that it is finding different classes of bodies. It could also be true that cadence is detecting structures that are not meaningful. Understanding these possible substructures would be very worthwhile in the future.

| Cluster | Abstracts |
|---------|-----------|
| 1 | 254 |
| 2 | 223 |
| 3 | 263 |
| 4 | 260 |

Chi-squared Statistic: 4.06

p-value (3 degrees of freedom): ~0.25

Table 3: k-means clustering of 1000 cadences representing abstracts.

| Cluster | Introductions |
|---------|---------------|
| 1 | 237 |
| 2 | 247 |
| 3 | 269 |
| 4 | 247 |

Chi-squared Statistic: 2.19

p-value (3 degrees of freedom): ~0.55

Table 4: k-means clustering of 1000 cadences representing introductions.

| Cluster | Bodies |
|---------|--------|
| 1 | 252 |
| 2 | 285 |
| 3 | 262 |
| 4 | 201 |

Chi-squared Statistic: 15.1

p-value (3 degrees of freedom): < 0.005

Table 5: k-means clustering of 1000 cadences representing bodies.

| Cluster | Conclusions |
|---------|-------------|
| 1 | 260 |
| 2 | 241 |
| 3 | 258 |
| 4 | 241 |

Chi-squared Statistic: 1.3

p-value (3 degrees of freedom): ~0.70

Table 6: k-means clustering of 1000 cadences representing conclusions.

4.4 Relative Contributions to Cadence

There are four factors that contribute to the cadence of a text segment. The first is the raw *tf-idf* weights and the other three are the heuristic operators. Experiments were performed to investigate the contributions of these factors individually and in combination with one another.

Table 7 through Table 9 show the results of these experiments using the same k-means clustering algorithm and the same 4000 test segments used in Section 4.2. The global precision is reported to help in comparing different experiments. Table 7 shows cluster results when the text segments are represented by their $f_i(x)$, functions which are the transformation of their raw *tf-idf* vectors onto the real line.

The low p-value suggests that these functions do indeed carry information useful in differentiating logical structures. It seems as though whatever this information is, it is comparably more muted and thus less discriminating than with cadence (see Table 2). This can be seen by the lower global precision of 0.511 ± 0.013 in Table 7 as compared to 0.822 ± 0.013 in Table 2. Closer examination of the clustering results shows marked decrease in precision and recall performances across the board. It is peculiar that introductions suffer the most performance degradation and that they are often misidentified as abstracts or conclusions. Perhaps there is information intrinsic to introductions that are brought forth to a greater extent by the heuristic operators than with other logical structures.

Table 8 shows clustering results when the k-means algorithm is applied to the functions $f_i(x)$ after they have been operated on by the coherence operator. Specifically, 4000 functions of the form $H_{i1}(f_i(x)) = f_i(x) - T_c \overline{f c}_i(x)$ are randomly assigned to four unlabeled clusters. The k-means clustering algorithm is then applied. Performance for introductions sees the most improvement by the coherence operator, followed by performance gains by abstracts and then conclusions. Performance on body segments was minimally affected. Further performance improvements are seen when the coherence operator is applied in conjunction with either the author or location operators (see Table 10 and Table 9).

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|-------------------|--------------|---------------|--------------|--------------|
| 1 | 636 | 354 | 83 | 177 |
| 2 | 283 | 224 | 146 | 153 |
| 3 | 7 | 93 | 584 | 72 |
| 4 | 74 | 329 | 187 | 598 |
| Precision | 0.509 | 0.278 | 0.772 | 0.503 |
| Recall | 0.636 | 0.224 | 0.584 | 0.598 |
| Balanced-F | 0.565 | 0.248 | 0.665 | 0.546 |

Global Precision: 0.511 ± 0.013

Chi-squared Statistic: 2417

Chi-squared p-value (15 degrees of freedom): less than 0.005

Table 7: Clustering results when using only the raw tf:idf vectors transformed onto the real interval [0, 1].

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|-------------------|--------------|---------------|--------------|--------------|
| 1 | 856 | 69 | 17 | 19 |
| 2 | 108 | 583 | 143 | 145 |
| 3 | 5 | 125 | 677 | 70 |
| 4 | 31 | 223 | 163 | 766 |
| Precision | 0.891 | 0.596 | 0.772 | 0.648 |
| Recall | 0.856 | 0.583 | 0.677 | 0.766 |
| Balanced-F | 0.873 | 0.589 | 0.721 | 0.702 |

Global Precision: 0.721 ± 0.013

Chi-squared Statistic: 5096

Chi-squared p-value (15 degrees of freedom): less than 0.005

Table 10: Only the coherence and authorship operators applied to the raw tf:idf vectors and then transformed onto the real interval [0, 1].

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|-------------------|--------------|---------------|--------------|--------------|
| 1 | 771 | 116 | 39 | 45 |
| 2 | 134 | 499 | 215 | 103 |
| 3 | 51 | 102 | 588 | 120 |
| 4 | 44 | 283 | 158 | 732 |
| Precision | 0.794 | 0.525 | 0.683 | 0.601 |
| Recall | 0.771 | 0.499 | 0.588 | 0.732 |
| Balanced-F | 0.782 | 0.512 | 0.632 | 0.660 |

Global Precision: 0.648 ± 0.013

Chi-squared Statistic: 3805

Chi-squared p-value (15 degrees of freedom): less than 0.005

Table 8: Only the coherence operator applied to tf:idf vectors and then transformed onto the real interval [0, 1].

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|-------------------|--------------|---------------|--------------|--------------|
| 1 | 917 | 66 | 44 | 24 |
| 2 | 60 | 605 | 141 | 119 |
| 3 | 1 | 141 | 576 | 63 |
| 4 | 22 | 188 | 239 | 794 |
| Precision | 0.873 | 0.654 | 0.738 | 0.639 |
| Recall | 0.917 | 0.605 | 0.576 | 0.794 |
| Balanced-F | 0.894 | 0.629 | 0.647 | 0.708 |

Global Precision: 0.723 ± 0.013

Chi-squared Statistic: 5322

Chi-squared p-value (15 degrees of freedom): less than 0.005

Table 9: Only the coherence and location operators applied to raw tf:idf vectors then transformed onto the real interval [0, 1].

4.5 Empirically Derived Aspects of Cadence

There are two aspects of cadence that have been observed. The first aspect was alluded to at the end of Section 4.1 as a phenomenon whereby the ability of cadence to differentiate logical structures is dependent on the corpus size. This phenomenon is termed the *Rule of 5000*. The second aspect has to do with the fact that the cadence of a logical structure seems to be invariant with topic domains. Each of these aspects is discussed below.

4.5.1 Rule of 5000

Recall the definition of cadence presented in Section 3.2. The cadence of a text segment is in part a function of the tf-idf weights of the non-stop terms in the segment. Each tf-idf is, in turn, a function of corpus size as shown in equation (8). Figure 6 shows a plot of global precision in clustering the same 4000 test segments used in previous experiments for different corpus sizes. Global precision performance appears to be adversely affected when corpus size is less than 5000. Figure 7 shows results from a different set of experiments examining the effects of corpus sizes under 4000. These experiments used 400 randomly selected text segments representing equally abstracts, introductions, bodies, and conclusions.

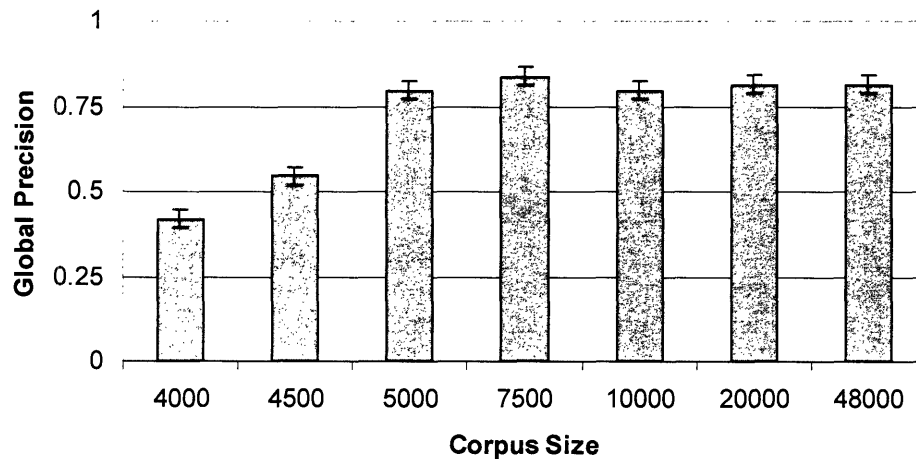


Figure 6: Experimental results showing global precision of k-means clustering algorithm at different corpus sizes. Test set comprised 4000 segments equally representing abstracts, introductions, bodies, and conclusions. Error bars represent ± 0.026 .

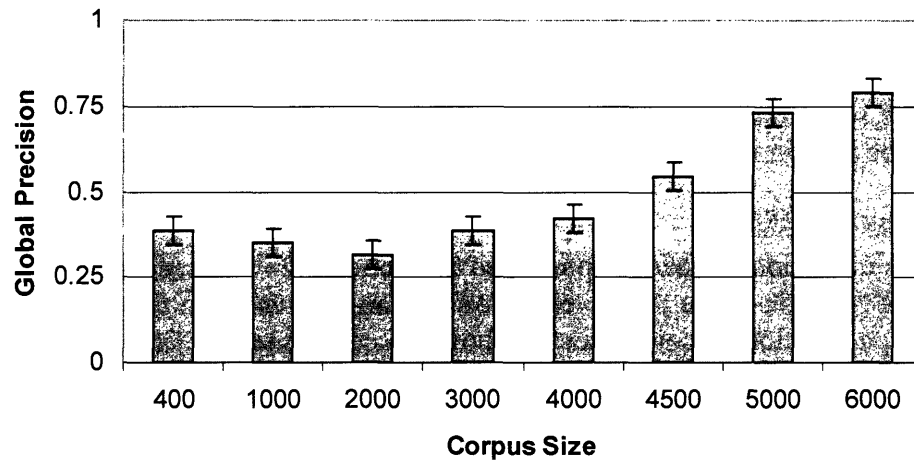


Figure 7: Experimental results showing global precision of k-means clustering algorithm at different corpus sizes. Test set comprised 400 segments equally representing abstracts, introductions, bodies, and conclusions. Error bars represents ± 0.041 .

One possible explanation for the poor performance with corpus size less than 5000 could be the under sampling of term frequencies.

4.5.2 The Topic Invariance of Cadence

When corpus size is kept at 48,000 documents, both of the test sets of 400 and 4000 segments show cadence plots similar to those shown in Figure 5. The clustering performances are also statistically equivalent to those presented in Table 2. The distribution of topics covered in each test set is presented in Table 11 below:

| Source | Topic | Test Set 400 | Test Set 4000 |
|--------------|------------------------|--------------|---------------|
| Cranfield | Aerodynamics | 3.5% | 4.1% |
| Medlars | Medicine | 3.3% | 2.9% |
| Times | World News | 12.4% | 13.2% |
| EBSCO | IT Technical Support | 18.5% | 19.3% |
| EBSCO | News & Popular Culture | 29.2% | 26.1% |
| EBSCO | Science & Medicine | 18.7% | 23.7% |
| EBSCO | Other | 14.4% | 10.7% |
| Total | | 100% | 100% |

Table 11: Distribution of topics in each of the test sets.

Contributions to each test set come predominately from the EBSCO collection because this collection is the largest subset within the annotated corpus. Within the EBSCO collection, both technical and non-technical topics from diverse areas are represented. Because clustering results are uniformly good at differentiating abstracts, introductions, bodies, and conclusions in all of the topics of Table 11, one might conclude that cadence is topic invariant. That is, the information carried in cadence that is useful to differentiating logical structures exists across the multiple topics. To understand a possible mechanism that may be responsible for this, it is instructive to look at the similarities and differences between cadence and the popular Salton vector-space formulation. Table 12 shows results from k-means clustering for the 4000 test segments where each segment is represented by its Salton vector. For the k^{th} segment, a Salton vector S_k of the form

$$\bar{S}_k = (tf_{k1} \cdot idf_1, tf_{k2} \cdot idf_2, \dots, tf_{kn} \cdot idf_n) \quad (9)$$

can be constructed where n is the number of terms in the vocabulary of the entire corpus and $tf_{kj} \cdot idf_j$ is term weight of the j^{th} term in the vocabulary. If a segment does not have a particular term, the corresponding vector element is zero. The vocabulary consists of an ordered set of non-stop terms designated as types. The ordering scheme does not matter, and the identity of each weight in the vector S_k is preserved by the subscript j . For the corpus size of 48,000 documents, there are 263,617 unique types in the vocabulary. When these types are combined to form proper nouns and proper phrases, the vocabulary grows to 4,563,205 types representing both terms and phrases*. This thesis uses the larger vocabulary. The vectors S_k are sparse by nature since any individual segment uses only a small fraction of types in the vocabulary.

The low p-value of Table 12 indicates that Salton vectors do carry information that prevents completely random grouping of segments. The clusters, however, do not correlate well with logical structures as indicated by the low precision, recall and balanced F scores.

* These terms and phrases are identified in the corpus by matching against a set of over 5 million terms and phrases in the Library of Congress Subject Headings.

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|-------------------|--------------|---------------|--------------|--------------|
| 1 | 201 | 183 | 194 | 173 |
| 2 | 241 | 247 | 240 | 245 |
| 3 | 266 | 257 | 275 | 248 |
| 4 | 292 | 313 | 291 | 334 |
| Precision | 0.268 | 0.254 | 0.263 | 0.272 |
| Recall | 0.201 | 0.247 | 0.275 | 0.334 |
| Balanced-F | 0.230 | 0.250 | 0.269 | 0.300 |

Global Precision: 0.264 ± 0.013

Chi-squared Statistic: 126

Chi-squared p-value (15 degrees of freedom): less than 0.005

Table 12: k-means clustering with parameter k=4 for 4000 segments where each is represented its Salton vector (transformed by equation 10).

When a similar experiment was performed using Salton vectors of 100 segments from a narrow topic domain, different performance results were obtained and a clearer picture emerged. The experiment used 25 abstracts, 25 introductions, 25 bodies, and 25 conclusions from documents in EBSCO collection on endothelial cell biology published between 1995 and 1999. Results from clustering these segments by their respective Salton vectors are shown in Table 13. With the exception of the introductions, all performance metrics improved when the topic domain is narrowed.

The discrepancies in performance between Table 12 and Table 13 are most likely due to the limitations of the Salton vector framework. This framework represents text segments that are similar in topics as vectors S_k that span similar subspaces. Likewise, text segments that do not address similar topics tend to use different types from the vocabulary, and so their vectors are made to span different subspaces. So, two text segments that represent the same logical structure will span very different subspaces if they address different topics. The k-means clustering results seen in both of the tables relies on having similar logical structures sharing similar vectors. One would therefore expect better performance for segments coming from a narrow topic domain if each segment is to be represented via Salton Vectors. One would also expect very poor performance when diverse topics are addressed in the test set. Results from Table 12 and Table 13 do indeed validate this line of thinking.

The reason cadence is topic invariant while Salton vectors are topic variant may be due to the fact that cadence is not dependent on the notion that “different topics should equal different subspaces”. In fact, there is no such notion at all in cadence.

It is interesting to note that the performance in Table 13 is statistically similar to that presented in Table 7. It is tempting to infer from this similarity that cadence is equivalent

to Salton vectors when Salton vectors are applied to a narrow topic domain. This is highly unlikely to be the case because Salton vectors are dependent on the ordering of the types in the vocabulary while cadence is dependent on the ordering of the words in a segment of text. In theory, the ordering of the terms in the vocabulary carries no information because retrieval results using Salton’s framework have been shown to be independent of ordering schemes. For cadence, however, it is likely that word ordering in a segment plays a pivotal role in helping to identify its logical structure.

The arguments presented above for why cadence is topic invariant is not sufficiently convincing to stand on its own. It relies heavily on the ability to attribute a cause for why Salton vectors are not topic invariant and then suggesting that since cadence is not susceptible to this same cause, cadence is thus topic invariant. This argument is purely conjecture, and ordinarily, one would be expected to dismiss such a conjecture if not for the overwhelming empirical evidence presented throughout this section and summarized in Table 2.

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|-------------------|--------------|---------------|--------------|--------------|
| 1 | 12 | 10 | 3 | 3 |
| 2 | 11 | 5 | 2 | 5 |
| 3 | 0 | 2 | 14 | 3 |
| 4 | 2 | 8 | 6 | 14 |
| Precision | 0.429 | 0.217 | 0.737 | 0.467 |
| Recall | 0.480 | 0.200 | 0.560 | 0.560 |
| Balanced-F | 0.453 | 0.208 | 0.636 | 0.509 |

Global Precision: 0.450 ± 0.165

Chi-squared Statistic: 51

Chi-squared p-value (15 degrees of freedom): less than 0.005

Table 13: k-means clustering with parameter k=4 for 100 segments where each is represented by equation 10. The 100 segments come from documents on endothelial cell biology published between 1995 and 1999.

4.6 Experimental Results with Entropy

The experimental evidence presented thus far for the utility of cadence in differentiating logical structures has assumed the *tf-idf* term weighting scheme. Another possible scheme is to weight terms based on their entropy and utility. This section presents the term weighting scheme used by Dumais [42] and results of hypotheses testing using this scheme.

In the *tfidf* metric, a term's weight in a particular document is in part a function of the number of times it occurs in that document. Therefore, a particular term – such as ‘cat’ – may have multiple *tfidf* scores, one for each document it appears in. Likewise, the entropy-based metric assigns multiple scores to each term in the entire document corpus.

For the z^{th} term in the j^{th} document that is part of a corpus of D documents, a term weight w_{zj} can be assigned by first computing utility scores for the term as follows

$$u_{zj} = \log(1 + f_{zj})$$

where u_{zj} is the utility score of the z^{th} term in the j^{th} document, and f_{zj} is the frequency of the z^{th} term in the j^{th} document. In a more general formulation, the term f_{zj} can be divided by a constant to shift the utility curve. For this thesis, we use a value of 1 for that constant. The utility function $\log(1+f_{zj})$ was arbitrarily chosen from among many other possible functions that also exhibit the characteristic of ‘diminishing return’. That is, presumably the 100th time a term is used in a document is marginally less important than the first few times it is used.

In addition to a utility score, one also computes an entropy score for each term as

$$H_z = -\sum_{j=1}^D p_{zj} \log(p_{zj})$$

where

$$p_{zj} = \frac{f_{zj}}{\sum_{j=1}^D f_{zj}}$$

p_{zj} is the normalized frequency of the z^{th} term in the j^{th} document. From the utility and entropy scores, one can compute the weight of the z^{th} term in the j^{th} document as

$$w_{zj} = u_{zj} \left(1 - \frac{H_z}{\log(D)} \right)$$

As written above, it should be noted that the quantity above is analogous to the *tfidf* metric in that u_{zj} is analogous to *tf* and $\left(1 - \frac{H_z}{\log(D)} \right)$ is analogous to *idf*.

To test the effects of using the entropy-based metric prescribed above in the computation of cadence, experiments similar to that of Section 4.2 were performed. Specifically, 1000 abstracts, introductions, bodies and conclusions were randomly selected from the annotated corpus for a total of 4000 segments. For each segment, its cadence is computed using the entropy-based metric prescribed above instead of *tfidf*. The weight of each term is also computed with respect to the larger 48,000 document corpus. A k-means clustering algorithm with k=4 was then applied. Table 14 presents the results.

The results of Table 14 suggest that using the entropy-based metric is statistically equivalent to using *tfidf* (compare with Table 2). During experimentation, however, it was noted that these results were volatile in that they depended on which segments were selected from the annotated test corpus. Specifically, when the experiment was repeated

| Cluster | Abstracts | Introductions | Bodies | Conclusions |
|---|----------------------|----------------------|----------------------|----------------------|
| 1 | 970 | 65 | 12 | 10 |
| 2 | 21 | 715 | 156 | 84 |
| 3 | 0 | 49 | 678 | 45 |
| 4 | 9 | 171 | 154 | 861 |
| Precision | 0.918 ± 0.025 | 0.733 ± 0.025 | 0.878 ± 0.025 | 0.721 ± 0.025 |
| Recall | 0.970 ± 0.026 | 0.715 ± 0.026 | 0.678 ± 0.026 | 0.861 ± 0.026 |
| Balanced-F | 0.969 | 0.749 | 0.791 | 0.810 |
| Global Precision: 0.806 ± 0.013 | | | | |
| Chi-squared Statistic: 6988 | | | | |
| Chi-squared p-value (15 degrees of freedom): less than 0.005 | | | | |

Table 14: k-means clustering results with parameter k=4.

for different sets of randomly selected segments, about two out every seven sets yielded poor precision and recall. The mechanism underlying this volatility is not completely clear, but the leading hypothesis suggests under-sampling. Recall that the entropy-based metric relies on the entropy of each term. A term's entropy in turn is a function of the term's normalized frequency distribution over the entire corpus of 48,000 documents. It is plausible that in some cases, one randomly selects segments that have term distributions that are not well characterized by the corpus-wide distributions. In these cases, one would expect that entropy would not be a good term weighting scheme, at least not for large corpora. This line of thinking has been partially corroborated by results of experiments where the entropy for each term is computed relative to a smaller corpus. These experiments yield results similar to those in Figure 7 (when *tfidf* is used with smaller corpus sizes) and they do not exhibit any noticeable volatility. It is interesting to note that the entropy-based metric is also subject to the Rule of 5000.

It appears that one can make several conclusions regarding the use of the entropy-based metric in the computation of cadence. First, when the entropy-based metric relies on term distributions that are representative of the segments being differentiated, the concept of cadence can be generalized to using either *tfidf* or entropy without loss of precision or recall performance. Second, both the entropy-base metric and the *tfidf* schemes for computing cadence are subject to the Rule of 5000. Third, with the caveat that this conclusion has the least amount of data to back it, the *tfidf* scheme may be the more robust of the two schemes for computing cadence since it does not appear to exhibit volatility for large corpora.

Section 5 Future Work

This thesis has thus far advanced evidence suggesting that cadence can be used to characterize text segments, and that cadence alone can differentiate segments representing different logical structures. The question remains as to whether cadence can be used to discover logical structures within documents. These two problems are related, but they are very different.

The first problem tests the ability to classify segments that are known to be abstracts, introductions, bodies or conclusions as one of these structures. The second problem requires the ability to demarcate where these structures begin and end within documents. For any given document, one must consider all possible segments and classify each one as one of the four structures or none of the four. For future work, the logical next step would be to use a supervised learning method to do exactly this.

One could use a k-nearest neighbor learning algorithm. It would require training data containing positive examples of cadences for each of the four structures and sufficient negative examples. At the very least, it would also require choices for k (the number of nearest neighbors), the size of the training set in terms of positive and negative examples, a scoring function, and the thresholds for classification based on a query's score.

The author suspects that the main obstacle with this and similar approaches will be developing a robust training set. Consider, for a moment, what will be necessary to demarcate logical structures accurately. Given a query document that may or may not contain one or more of the four logical structures, demarcating logical structures will require precision at the word or line level; the first word (or line) would be the beginning of the structure and the second would be the ending*. In either case, finding these demarcations will rely on the resolving power of cadence to detect changes at relatively small scales relative to the structure sizes themselves. That is, for example, the average number of lines in the smallest of the structures, the abstract, is sixteen. It stands to follow then that the inclusion or exclusion of a particular line during the demarcation process would result in less than $1/16$ change in the overall shape of the cadence of a candidate segment.

* Using higher order structures, such as paragraphs, are not always feasible because in the parsing of large numbers of diverse documents, these markers between these larger structures are usually lost.

With such small changes, one could imagine that the cadences of many possible text segments would look close enough to each other and to that of the true abstract that the k-nearest neighborhood algorithm would fail to reliably pick out the correct one. This is the best of the scenarios. For bodies or conclusions where the sizes of structures – as measured by number of lines or words – is several orders of magnitude larger, the ability to pick out the true demarcations from nearby candidates is even more daunting.

One way to view this difficulty is in terms of length-scales. It is clear that the longer a text segment is, the less important smaller-scale phenomena at the word or line level become relative to larger phenomena at the multi-line or page level. It is also clear that these smaller-scale phenomena are the ones most likely to contain the information to accurately demarcate down to the correct word or line.

How can one use the cadences of whole logical structures to see both small and large-scale phenomena? The author suspects that possible solutions lie in what was alluded to in Section 4.3 . Perhaps larger logical structures contain logical substructures. If these are present, and if the training set contains sufficient positive examples of them, perhaps they may provide the necessary smaller-scale phenomena that one needs. For not-so-large logical structures, it is most likely that logical substructures do not exist, but perhaps there may be smaller-scale non-logical substructures that one can take advantage of.

These statements are a testament to the future work that will be required as they are littered with ‘perhaps’.

Section 6 Concluding Remarks

This thesis concludes with remarks on where cadence owes its roots. The use of cadence to characterize text is merely a new interpretation of a larger idea put forth by Gerard Salton more than two decades ago. Like Salton's method, cadence is based on vectors and vector arithmetic. The two methods each enable in their own right separate families of functions on text where members of a family differ by differing term-weighting schemes. Both methods are founded on the notion that similarities and differences between their vector representations are somehow important. These are the most obvious similarities.

The differences between the two methods are subtle. Both methods use vectors of the form $(w_{i1}, w_{i2}, \dots, w_{im_i})$. The m_i subscript is a constant for all vectors in Salton's method, but it is a property of individual segments in cadence. The elements w_{ij} has the j subscript referring to an index in a corpus-wide vocabulary in Salton's method, while cadence has this same subscript referring to the order in which a word appears in a text. If we ignore for now the three heuristics, these two differences hardly cry out as glaring differences. They are, after all, differences at the subscript level.

In hindsight, Salton vectors and cadence are very different in their intentions. Salton vectors were designed to compare text segments and their topics. They rely on the identity of words and their frequencies. They assume that topically similar documents intrinsically use similar words. They rely on little else, and so they pay little attention to word order. Their simplicity and elegance has been shown to be very effective at what they try to do. When Salton vectors are used to differentiate logical structures, they perform miserably. It is as if they fail to capture whatever it is that humans can read and see as intrinsically there. The evidence in this thesis suggests that cadence may indeed be that intrinsic property.

Section 7 Bibliography

1. K. Summers, Automatic Discovery of Logical Document Structure, Ph.D. Thesis, Cornell University, August 1998.
2. S. Mao, A. Rosenfeld, T. Kanungo, Document structure analysis algorithms: a literature survey, Proc. SPIE Electronic Imaging, Vol. 5010, pp. 197-207, January 2003.
3. R. Haralick, Document Image Understanding: Geometric and Logical Layout, Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, Seattle, 1994.
4. G. Nagy, Twenty years of document image analysis in PAMI," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, pp. 38-62, 2000.
5. H. Fujisawa, Y. Nakano, and K. Kurino, Segmentation methods for character recognition: from segmentation to document structure analysis, Proceedings of the IEEE, Vol. 80, pp. 1079-1092, 1992.
6. S. Tsujimoto and H. Asada, Understanding multi-articled documents, Proceedings of International Conference on Pattern Recognition, Atlantic City, NJ, pp. 551-556, June 1990.
7. A. Yamashita, T. Amano, I. Takahashi, and K. Toyokawa, "A model based layout understanding method for the document recognition system," in Proceedings of International Conference on Document Analysis and Recognition, Saint-Malo, France, pp. 130-138, September 1991.
8. J. L. Fisher, Logical structure descriptions of segmented document images, Proceedings of International Conference on Document Analysis and Recognition, Saint-Malo, France, pp. 302-310, September 1991.
9. F. Wahl, K. Wong, and R. Casey, Block segmentation and text extraction in mixed text/image documents, Graphical Models and Image Processing, Vol. 20, pp. 375-390, 1982.
10. L. A. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, pp. 910-918, 1988.
11. L. O'Gorman, The document spectrum for page layout analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, pp. 1162-1173, 1993.
12. K. Kise, A. Sato, and M. Iwata, Segmentation of page images using the area Voronoi diagram, Computer Vision and Image Understanding, Vol. 70, pp. 370-382, 1998.

13. A. K. Jain and B. Yu, Document representation and its application to page decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, pp. 294-308, 1998.
14. G. Nagy, S. Seth, and M. Viswanathan, A prototype document image analysis system for technical journals, *Computer*, Vol. 25, pp. 10-22, 1992.
15. H. S. Baird, S. E. Jones, and S. J. Fortune, Image segmentation by shape-directed covers, *Proceedings of International Conference on Pattern Recognition*, Atlantic City, NJ, pp. 820-825, June 1990.
16. G. E. Kopec and P. A. Chou, Document image decoding using Markov source models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 602-617, 1994.
17. T. A. Tokuyasu and P. A. Chou, Turbo recognition: a statistical approach to layout analysis, *Proceedings of SPIE Conference on Document Recognition and Retrieval*, San Jose, CA, January 2001.
18. T. Kanungo, S. Mao, Stochastic language models for style-directed layout analysis of document images, *IEEE Transactions on Image Processing*, Vol. 12, No. 5, pp. 583-596, 2003.
19. S. Mao and T. Kanungo, Empirical performance evaluation methodology and its application to page segmentation algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, pp. 242-256, 2001.
20. G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, Vol. 18, pp. 613-620, 1975.
21. H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, Vol. 1, No. 4, pp. 309-317, 1957.
22. K. Sparck Jones, A Statistical Interpretation of Term Specificity and Its Application in Retrieval, in *Journal of Documentation*, Vol. 28, No. 1, pp. 11-20, 1972.
23. G. Salton. Experiments in automatic thesaurus construction for information retrieval. *Proceedings of the IFIP Congress*, Ljubljana, YU, Vol. TA-2, pp. 43-49, 1971.
24. G. Salton, C.S. Yang, and C.T. Yu, A theory of term importance in automatic text analysis, *Journal of the American Society for Information Science*, Vol. 26, pp. 33-44, 1975.
25. Y. Tateisi and N. Itoh, Using stochastic syntactic analysis for extracting a logical structure from a document image, *Proceedings of International Conference on Pattern Recognition*, Jerusalem, Israel, pp. 391-394, October 1994.
26. R. Brugger, A. Zramdini, and R. Ingold, Modeling documents for structure recognition using generalized n-gram, *Proceedings of International Conference on Document Analysis and Recognition*, Ulm, Germany, pp. 56-60, August 1997.
27. A. Dengel and F. Dubiel, Computer understanding of document structure, *International Journal of Imaging Systems and Technology*, Vol. 7, pp. 271-278, 1996.
28. H. Ahonen, Generating grammars for structured documents using grammatical inference methods. Report A-1996-4, Department of Computer Science, University of Finland, 1996.
29. M. D. Young-Lai, Application of a stochastic grammatical inference method to text structure, Master's thesis, Computer Science Department, University of Waterloo, 1996.

30. J. S. Key, R. K. Wong, Structural inference for semistructured data, Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management (CIKM), Eds. Calton Pu; David Grossman; Ling Liu, Atlanta USA, 5-10, pp. 159-166, Nov. 2001,
31. T. A Bayer, H. Walischewski, Experiments on extracting structural information from paper documents using syntactic pattern analysis, Proceedings of the 3rd International Conference on Document Analysis and Recognition, pp. 476-479, August 1995.
32. C. Wenzel, Supporting information extraction from printed documents by lexicon-semantic pattern matching, Proceedings of the 4th International Conference on Document Analysis and Recognition, pp. 732-739, August 1997.
33. T. Saitoh, M. Tachikawa, and T. Yamaai, Document image segmentation and text area ordering, Proceedings of International Conference on Document Analysis and Recognition, Tsukuba Science City, Japan, pp. 323-329, October 1993.
34. D. Niyogi, S. N. Srihari, Knowledge-based derivation of document logical structure, Proceedings of Inter-national Conference on Document Analysis and Recognition, Montreal, Canada, pp. 472-475, August 1995.
35. G. Salton , A. Singhal , C. Buckley , M. Mitra, Automatic text decomposition using text segments and text themes, Proceedings of the seventh ACM conference on Hypertext, Bethesda, Maryland, United States, pp.53-65, March 16-20, 1996.
36. C. C. Lin, Y. Niwa, and S. Narita, Logical structure analysis of book document images using contents information, Proceedings of International Conference on Document Analysis and Recognition, Ulm, Germany, pp. 1048-1054, August 1997.
37. T. Kochi, T. Saitoh, User-defined Template for Identifying Document Type and Extracting Information from Documents, Proceedings of the 5th International Conference on Document Analysis and Recognition, pp. 127-130, September 1999.
38. Y. Ishitani, Logical structure analysis of document images based on emergent computation, Proceedings of International Conference on Document Analysis and Recognition, Bangalore, India, pp. 189-192, September 1999.
39. G. Salton, The SMART System --- Experiments in Automatic Document Processing, Prentice Hall, 1971.
40. C. W. Cleverdon, J. Millis, E. M. Keen, Factor determining the performance of indexing systems, 2 vols. Cranfield, 1966.
41. F. W. Lancaster, Information retrieval systems: characteristics, testing and evaluation, New York, Wiley, 1968.
42. S. T. Dumais, Improving the retrieval of information from external sources. Behavior Research Methods, Instruments and Computers, 23(2), 229-236, 1991.