Title
A Cognition-Driven Approach To Modeling Document Generation and Learning Underlying Contexts From Documents

Author
Falahi, Misagh

Publication Date
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

A Cognition-Driven Approach To Modeling Document Generation and Learning Underlying Contexts From Documents

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Misagh Falahi

2017

ABSTRACT OF THE DISSERTATION

A Cognition-Driven Approach To Modeling Document Generation and Learning Underlying Contexts From Documents

by

Misagh Falahi
Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2017
Professor Vwani P. Roychowdhury, Chair

The development of the Web has, among its other direct influences, provided a vast amount of data to researchers in several disciplines. While in the early stages of its growth the data often went unseen and was secondary to the other products the Internet made available, in the past decade it has quickly become a primary resource for a large number of online applications and has given possibility to many analyses and studies. Text data in particular has been a cornerstone of these works in an attempt to better understand human knowledge and behavior.

This work focuses on analysis of the process of writing documents and the abstract underlying contexts driving this process. We propose a generative model for documents based on psychological models of human memory search, and from there we define structures that can represent these abstract contexts.

Recent works in psychology literature suggest the brain's memory search can be modeled as a random walk on a semantic network (Abbott et al., 2012). The vast body of research available on random walks in different disciplines, and more recently for their use in analyzing the structure of the web and developing search engines, makes this model particularly appealing for understanding and simulating the brain's process of vocabulary selection and document generation. It can also be used to drive lexical applications and automated text analyses such as exploring the inherent structures existing in a language and the relationship between words.

In this work, we present a network approach to describing document generation and discover-

ing contexts. We form an associative network of words based on co-occurrence, with ties between words weighted by the number of documents in the corpus they simultaneously appear in. By inspecting the hierarchical modularity of this network and using the random walk model and community detection algorithms based on random walks, we can find communities of words that form contextually homogeneous groups. Within a certain context defined by one of these groups, the relative importance of every other word can be determined by creating a contextually biased word association network and using the Google PageRank algorithm that magnifies nodes with higher centrality. We use these context profiles to form a context-term matrix representative of semantic traces in memory. We then study the hierarchical structure of contextually significant word clusters in different layers of the network, through examining layer blocks of the context-term matrix.

Other similar studies include topic modeling, the unsupervised learning of patterns of words and phrases that can represent "topics". The mainstream view in topic modeling regards a topic as a distribution over known vocabulary. The famous Latent Dirichlet allocation (LDA) for instance (Blei et al., 2003), finds a given number of topics within a text corpus, each topic represented by a distribution over all words. LDA essentially fits a latent variable model of word combinations to a set of observed documents.

We also extend our knowledge structure model to find vector representations of topics that provide summaries of the information contained in the corpus, similar to topic modeling frameworks. These vector representations are calculated by factorization of the context-term matrix. The summary outcome of this method will also reveal important sub-structures of the large hierarchical structure. For evaluation, we show that across a variety of datasets from online forums and tweets to research articles, our summary topics cover, on the average, 94% of $k = 60$ LDA topics.

The dissertation of Misagh Falahi is approved.

Lieven Vandenberghe

Mark S. Handcock

Timothy R. Tangherlini

Vwani P. Roychowdhury, Committee Chair

University of California, Los Angeles

2017

To my parents, who taught me to value knowledge.

To my siblings and my friends, for their support, be it emotional or material.

And to those who taught me music, without whom it would have been a silent journey . . .

All that you touch

And all that you see

All that you taste

All you feel . . .

And all you create

And all you destroy

And all that you do

And all that you say . . .

And all that is now

And all that is gone

And all that's to come

And everything under the sun is in tune

But the sun is eclipsed by the moon.

– "Eclipse" by Roger Waters

TABLE OF CONTENTS

1 Introduction and Background **1**

1.1 Motivation 1

1.2 Psychology Background 2

1.3 Outline 4

2 Modeling Document Generation and Underlying Knowledge Structure **6**

2.1 Problem Statement 6

2.2 Related Work 6

2.2.1 Topic Modeling 6

2.2.2 LSA (Latent Semantic Analysis) 7

2.2.3 Mixture of Unigrams 7

2.2.4 pLSI (Probabilitstic Latent Semantic Indexing) 8

2.2.5 LDA (Latent Dirichlet Allocation) 8

2.3 Network Model 9

2.4 Network Structure 10

2.4.1 Google PageRank 11

2.4.2 Word Generality and Network Structure 11

2.5 Modeling Document Generation 13

2.5.1 Contextual Memory 14

3 Methodology **18**

3.1 Random Walk with Reflection 19

3.2 Hierarchical Clustering 20

3.3 Context Networks 24
3.4 Context Profiles 28
3.5 Concept Hierarchy 35
3.5.1 Extracting Concept Clusters In Different Layers 35
3.5.2 Hierarchical Structuring 44
3.6 Summarization 51
3.6.1 Summary Topics 51
3.6.2 Summary Structures 58
3.7 Document Mapping 63

4 Experimental Results and Evaluation 66
4.1 Datasets 66
4.2 Evaluation of Summary Topics 68
4.3 Evaluation of Layered Clusters 75

5 Discussion and Further Work 84
5.1 Discovering Structures from Data 84
5.2 Actant Clustering 88

6 Tools and Other Analyses 93
6.1 Community Detection 93
6.1.1 InfoMap 93
6.2 Non-Negative Matrix Factorization 94
6.3 Algorithm Complexity 96

7 Conclusion 100

References **102**

LIST OF FIGURES

1.1 Experimental Inter-Response Time Recorded by Hills et al. [6] 2

1.2 Simulated IRT using the random walk model by Abbott et al. [2] 4

2.1 Mixture of Unigrams model. Diagram from [3]. 7

2.2 pLSI (Probabilistic Latent Semantic Indexing) model. Diagram from [3]. 8

2.3 LDA (Latent Dirichlet Allocation) model. 9

2.4 LDA topic distribution samples 9

2.5 3 nodes in the network with respective edge weights 10

2.6 Complementary CDF in log scales for the NSF (left) and Cafemom (right) datasets. . . 12

2.7 A small sub-graph of the network showing a few general nodes that connect contexts . 13

2.8 A sample piece of a hierarchical structure for one of our datasets containing forums of parents discussing vaccination on a website named Mothering.com. This piece is about "exemption from vaccination" for children. These concepts arise from parents' determination to not vaccinate their children, and searching for strategies to avoid required vaccination, such as religious exemption laws. Average PageRank of nodes in each layer cluster words is noted on side. 17

3.1 Breakdown of steps to extract the hierarchical concept structure. 19

3.2 Random walk reflection probability as a function of PageRank difference. The shift in the sigmoid function is also a linear function of the destination PageRank. 20

3.3 Change in distribution of effective PageRank with modified random walk for the NSF (left) and Cafemom(right) datasets 21

3.4 For a bottom-layer cluster of words and a given outside word, the co-occurrence values between the outside word and words of the cluster carry information about how many times the outside word appears in the document set containing the cluster words. . . . 26

3.5 For a pair of words outside the core context cluster C we estimate their contextual association w_{ij}^C using the estimated contextual occurrence counts N_i^C, N_j^C 27

3.6 Context-to-global PageRank ratio plot (PageRank profile) of a context in NSF abstracts (top) and a context in vaccination discussion forums of Mothering.com (bottom). Each point is a word, with its position on the x-axis equal to the word's global PageRank (PageRank in global co-occurrence Network), and its position on the y-axis equal to the context-to-global PageRank ratio. The core cluster that defines the context gets the highest boost and can be seen in the points on the top left. Each context brings up some words at different PageRank ranges above its core. 29

3.7 Outliers in PageRank profile of a context in NSF abstracts on *"nanostructures"*. The outliers are colored green, purple, and yellow, in decreasing order of significance. This crop shows middle-layer bins 10,11,12 out of the 20 PageRank bins. 31

3.8 Bins 14,15,16 out of the PageRank profile in figure 3.7 32

3.9 PageRank profile of a context in Mothering.com forums discussing vaccination. This crop shows middle-layer bins 10,11,12 out of the 20 PageRank bins for a context about vaccine ingredients. 33

3.10 Bins 14,15,16 out of the PageRank profile in figure 3.9 34

3.11 Bin 12 of different PageRank profiles for three related contexts in the NSF abstracts dataset. The core cluster of contexts from left to right are:

1) qos congestion end-to-end ip packet tcp routers guarantees packets atm multicast node router admission buffer delays guaranteed routing scalability nodes guarantee destination traffic topologies delay protocol hoc

2) cmos interconnect on-chip interconnects ic layout vlsi chips mixed-signal low-power chip ics circuit analog circuits

3) fading multipath decoding cdma equalization multiuser turbo trellis coded wideband dsp reception cancellation shannon isi 35

3.12 Bins 15 and 16 of different PageRank profiles in the Mothering.com forums datasets for three related contexts about doctor visits. The core clusters of contexts from top to bottom are:

1) wbvs checkups check-ups wbc appts physicals neglectful cya practioner well-child receptionist harassed neglecting hassled ins referrals referral hmo pedis harass emergencies fp harassment

2) check-up height well-baby checkup cancel appointments weighed gaining exam respects ups checks pressured lecture skipped measured trail

3) practitioners respects chiro prescribe gp pocket fp well-baby checkup exam weighed cancel appointments skipped ups checks . 36

3.13 Bin 9 of three context PageRank profiles in the Mothering.com mentioned in figure 3.12. All three are about doctor visits, but mostly about finding doctors that are not judgemental of non-vaccinating parents. The first one also talks about scheduling office visits significantly. 37

3.14 Directional bin-wise overlap values for the first two Mothering.com contexts in figure 3.12. The respective indices of the two in dataset contexts are 66 and 30. Other sample contexts with no similarity to 30 are shown for comparison. 38

3.15 Directional bin-wise overlap values between the food-related context 27 in Mothering.com and two other contexts 29 and 35, about food and homeopathy. Other sample contexts with no similarity to 27 are shown for comparison. The core clusters for these contexts are:

27: supplements sugar oil vitamins drink organic store liver

29: lemon teaspoon capsules grams tastes tsp ascorbic olive tablets capsule gram hylands leaf flax emergen-c baking tabs colloidal acidophilus coconut iodine omega spinach herb endotoxin na silica

35: herbs remedy herbal homeopath medicines allopathic traditional remedies homeopathy homeopathic 39

3.16 Hierarchical structure of the layered NMF clusters extracted from forums of Mothering.com. Very small values of edge weights below 0.3 are filtered out for better visualization. 50

3.17 A crop of the hierarchical structure of NMF clusters showing some clusters on "vaccine ingredients" and "autism" in the four layers ranging from L_{12} (bottom nodes) on bins $(12, 13, 14)$ up to L_{15} (top nodes) on bins $(15, 16, 17)$. Top 5 words of each cluster are shown on its respective node. 50

3.18 A crop of the hierarchical structure of NMF clusters showing some clusters on "scheduling doctor appointments" and "homeopathy and alternative medicine" in the four layers ranging from L_{12} (bottom node) on bins $(12, 13, 14)$ up to L_{15} (top node) on bins $(15, 16, 17)$. 51

3.19 Row vectors of matrix W^L that contain a context's mapping onto layer clusters. The vectors for 6 different Mothering.com contexts are shown in middle layer L_{10} which spans over bins $(10, 11, 12)$. 59

3.20 Mapped structure of a context onto the concept hierarchy. Normalized mapping values $\hat{W}^L_{cv}$ from context c to layer clusters v are encoded in node colors, with darker colors assigned to higher values. A distant view of the mapped highlighted chain is shown on top, and on the bottom a close-up of some mapped layer clusters. This context is about food and home remedies, built around the following core cluster:

• lemon teaspoon ascorbic capsules grams tastes kefir tsp mgs gram omega emergen-c hylands leaf capsule absorption flax olive coconut kg iu tablets b12 diarrhoea baking dosages acidophilus herb spinach iodine tabs copper migraines cap cabinet folic 60

3.21 A distant view (left) and close-up (right) of the mapped structure of a summary topic onto the concept hierarchy. Darker node colors depict higher mapping coefficients α_v in the solution of the sparse regression problem. The topic, extracted from Mothering.com by setting $k = 60$ summary topics, is about the use of fetal cells in vaccine production and the top 10 words of the topic are:

• aborted fetal cells tissue fetus ingredients lines cell cultured monkey 62

3.22 A distant view (left) and close-up (right) of the mapped structure of an LDA topic onto the concept hierarchy. Darker colors show higher mapping coefficients. This topic is again on "fetal cells" extracted from Mothering.com forums, setting LDA number of topics $k = 20$. The top 10 words for this topic are:
• aborted fetal cells tissue cell dna human monkey abortion 63

3.23 A distant view (left) and close-up (right) of the mapped structure of a document onto the concept hierarchy. Darker colors show higher mapping coefficients. This document is a Mothering.com thread on "food and supplements":
... Elimination diets and definately not Rotation Diets like she suggests, by themselves do not heal the gut. And then, you have to supplement to replace all the food you are taking out (and supplements are not as effective as food based nutrients and also can be damaging to a leaky gut b/c they are artificial.) My DS has been on minerals and special diet since Sept. 64

3.24 A distant view (left) and close-up (right) of the mapped structure of a document onto the concept hierarchy. Darker colors show higher mapping coefficients. This document is a Mothering.com thread on "homeopathy":
... Lydall's book is more about alternative treatments vs lots of specific vax info. I found this book to be so poorly done... While I'm not really into homeopathy, her book is written from the perspective of someone who has actually treated these disease, so she's able to breakdown what to expect in a normal course of illness. 65

4.1 Mapping coefficients from LDA topics and summary NMF topics for four datasets. . . 70

4.2 Maximum mapping coefficient from each LDA topic to summary NMF results of four different runs on each of the four datasets. For all of the displayed plots $k = 60$ in both methods. 71

4.3 Mapping coefficients from LDA with $k = 60$ topics to summary NMF with $k = 100$ topics for Mothering.com threads. 72

4.4 Word coverage curves for top words of 6 Mothering.com LDA topics in their best-match summary NMF topic. Each point (i, j) drawn on a plot displays the smallest j for which 80% of top i words of the LDA topic are covered in the top j words of the summary NMF topic. 76

4.5 Z-score-weighted coverage of top 100 LDA topic words in top 100 words of NMF topics. Topic numbers are permuted to reduce the matrix bandwidth such that higher values tend to get closer the main diagonal [41]. 77

4.6 Histogram of documents' fraction of words from bottom-layer clusters for Mothering (left) and Cafemom (right). 78

4.7 Modularity of layer clusters for layers 7, 10, 13, and 17 on Mothering.com inside the layer's subgraph of co-occurrence network. The red line indicates standard deviation of modularity for 10 random nodes in the layer. The average value of modularity for randomly selected clusters is relatively very small compared to the standard deviation. 80

4.8 Average modularity of layer clusters (blue dots) and random clusters of same size in the same layer (red dots). Boxes on points display 90% confidence intervals. These values are shown for four datasets of Mothering.com threads (top left), Cafemom threads (top right), Iran Election Tweets (bottom left), NSF abstracts (bottom right). 81

4.9 Average modularity plots of layer clusters described in figure 4.8, for three datasets of Reuters news articles (top left), Children's Fiction (top right), and 18th Century Collections (bottom). 82

5.1 A crop of the extracted hierarchical structure of Mothering.com that relates to "exemption". One can see different branches of concepts about "religious exemption" , "required forms", "state laws", and "school regulations" merging at the higher layers to indicate their shared scope of getting exemption from vaccines to enroll children in school. 85

5.2 A crop of the extracted hierarchical structure of Cafemom showing chains related to "exemption requirements" and "state laws". 86

5.3 A close-up of the hierarchical structure of Cafemom concepts relating to "vaccine safety". The bottom plot shows a related chain in the middle layer and the top plot shows continuation of the same chain at the top layers that mixes with concepts on "reading and research about vaccines". 87

5.4 A distant view of the hierarchical structure extracted from 18th Century Collections. The plot is broken into three parts due to size; the 3 figures from top to bottom display the left, middle, and right parts of the hierarchy. The large variety in this collection and lack of a central scope, creates a very divided hierarchy of mostly independent chains. . 88

5.5 A close-up of two hierarchical structure pieces from the 18th Century Collections dataset. These pieces are related to "greek mythology" (left) and "writing" (right). . . . 89

5.6 Contextual Network Model of Stories. Nodes represent actants $A_1, ..., A_n$, and edges carry information about relationships and interactions that arise in a particular story context between pairs of actants. 90

LIST OF TABLES

3.1 Five examples of bottom-layer clusters extracted from a dataset of NSF funding abstracts. These clusters are related to relatively detailed contexts from "astronomy", "statistics", "education", "biology", "economics". 22

3.2 Four different observations of the same bottom-layer clusters extracted from the NSF funding abstracts dataset. These clusters are observed in community detection outcome for four different parameter settings. 23

3.3 Layer clusters 1 to 20 out of 50 clusters extracted using the NMF method on a window over PageRank bins (11,12,13) of Mothering.com dataset. 45

3.4 Layer clusters 21 to 40 out of 50 clusters extracted using the NMF method on a window over PageRank bins (11,12,13) of Mothering.com dataset. 46

3.5 Layer clusters 41 to 50 out of 50 clusters extracted using the NMF method on a window over PageRank bins (11,12,13) of Mothering.com dataset. 47

3.6 Summary topics extracted for forums of Mothering.com using the weighted NMF method on context PageRank profiles. The number of topics is set to $k = 20$. 55

3.7 Two topics related to interactions with doctors in $k = 60$ NMF topics. These topics are combined in topic 11 of table 3.6 when we extract $k = 20$ topics. 56

3.8 LDA topics extracted for Mothering.com forums with $k = 20$ topics. The words of each topic are ordered according to their z-score, instead of the actual probability values in multinomial distributions stored in rows of β. This ordering helps bring up more contextually important words of each topic, and suppresses more general words that naturally have high values in topic distributions that are trying to reconstruct distribution of words in documents. 57

4.1 Datasets Statistics: Mothering.com Threads, Cafemom Threads, NSF Abstracts, Iran Election Tweets. 67

4.2 Datasets Statistics: Reuters News Articles, 18th Century Collections Online (ECCO-TCP), Children's Fiction. 67

4.3 Number of LDA topics covered in various k summary NMF topics. For all datasets $k = 60$ LDA topics are analyzed. In some cases, number of covered topics stays constant or even decreases when going from $k = 80$ to $k = 100$ summary NMF topics. This is because for larger numbers topics are divided differently and thus some old topics are distributed across several new topics. By taking the number of LDA topics covered in at least one of these three different NMF summaries of each dataset, on average 94% of a dataset's $k = 60$ LDA topics are covered in the summary. 73

5.1 Actant name clusters extracted from context co-occurrence of recurring nouns in Mothering.com. Cluster 1 contains different names that refer to "doctors" and cluster 2 mainly contains "diseases". The second example cluster is quite large so only parts of it are shown here. 91

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor Professor Vwani Roychowdhury for his support and mentorship. Our long discussions always lit up the road ahead. I would also like to express my gratitude to the members of my doctoral committee, Professor Timothy Tangherlini, Professor Mark Handcock, and Professor Lieven Vandenberghe, for taking the time to serve on this committee and providing feedback on this work. I should also acknowledge the efforts of Zicong Zhou whose thesis work started this project.

I wanted to thank my sister and brother and my friends for all the help they have given me in various ways during these years. This would have not been possible without you. I also wanted to thank the crew at the Culver City coffee shop "Bar Nine" where much of this manuscript was written.

Finally, I would like to thank my parents for their immeasurable love and support. There is nothing I could say that would express how grateful I am for what you have done for me.

VITA

2012	B.S. (Electrical Engineering), Sharif University, Tehran, Iran.
2014	M.S. (Electrical Engineering), University of California, Los Angeles.
2012–2013	Research Assistant, Computer Science Department, University of California, Los Angeles.
2013–2015	Teaching Assistant, Electrical Engineering Department, University of California, Los Angeles.
2016–2017	Research Assistant, Electrical Engineering Department, University of California, Los Angeles.
2016	Culture Analytics Semester Program, Institute for Pure and Applied Mathematics (IPAM) at UCLA.

CHAPTER 1

Introduction and Background

1.1 Motivation

Understanding the structure of human knowledge through modeling the process of writing documents has been of great interest in several disciplines. Models of document generation have significant value in our understanding of the human brain and semantic memory [17]. For decades, computational psychologists have been designing experiments to analyze the human brain and how we search our memory when using language [5] [6]. Most of these studies focus on memory recollection and word association tasks, and devise models that describe experimental results from such tasks [2]. Meanwhile, computer scientists have taken a statistical approach to describe patterns of text observed in documents [3]. There is still a need to close the gap between these two schools of thought; to design models that statistically describe observed documents and also match models of memory.

We base our model of document generation on models of human memory [2] [18] and show how we can leverage large text data collected from the Web to train this model. Our model will be fairly intuitive and will use well-studied network analysis and matrix factorization tools to obtain a sequential solution to the model structure estimation problem. This sequential optimization approach to estimating the various layers of our model makes the final results more interpretable and further analysis easier. Methods for joint estimations of the parameters of our model are left as part of future research work. We perform a number of statistical evaluation tests to show that the sequential optimization approach already gives superior results.

1.2 Psychology Background

There have been considerable research in Psychology and Cognitive Sciences in an effort to understand and model how human memory works. One of the prevalent experiments, that more specifically investigates *semantic memory*, is the semantic fluency task. In a prominent case study, Troyer et al. [5] asked a group of subjects to retrieve examples of animals and then analyzed the clustering of words in the recorded responses. Their observation was that when asked for related words to a certain query, people recite different clusters of words in bursts. In the case of animals these clusters reflect sub-categories of animals like "pets", "African animals", etc. They concluded that semantic memory search can be decomposed into two separate "clustering" and "switching" processes.

Hills et al. [6] conducted a similar experiment in which they asked respondents to name as many animals as they could in 3 minutes and observed the Inter-item Response Time (IRT), which they defined as the time between consecutive words in the sequence generated by each subject. They compared the IRT with patches (clusters) found by Troyer et al. and found that there is a jump in IRT for consecutive words that are in different patches, corresponding to the "switching" process. In comparison, the "clustering" process implies relatively similar IRT for consecutive words that are in the same patch.

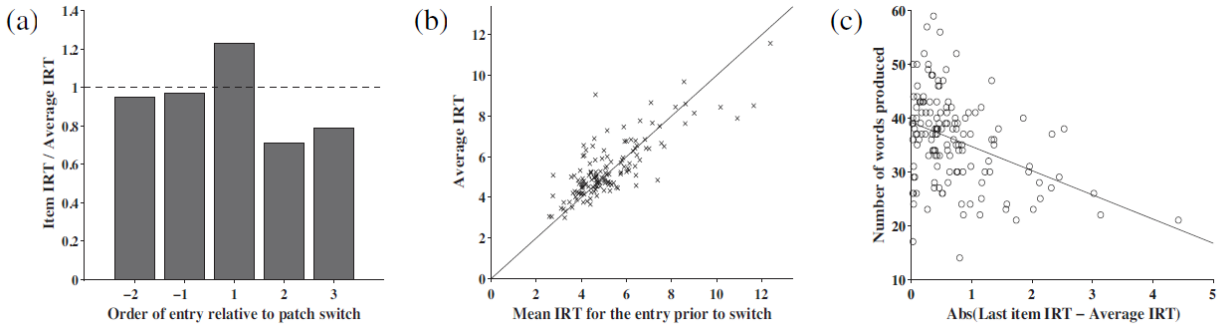


Figure 1.1: Experimental Inter-Response Time Recorded by Hills et al. [6]

Abbott et al. [2] later attempted to integrate "clustering" and "switching" into a single process by modeling memory search as random walks on a semantic network. Random walk models had also previously shown to reveal prominence of words in human memory [7]. In the model proposed

by Abbot et al., memory search is regarded as the position and movements of a random walker on a semantic network, assuming the walker starts from a cue word like "animal". The random walker will then randomly jump to one of the neighbor nodes. The probability of jumping to each neighbor is proportional to the weight of the associative link to it from the current node.

$$P(X_{n+1}|C = "animal", X_n = x_n) = \rho P(X_{n+1}|X_n = "animal") + (1-\rho)P(X_{n+1}|X_n = x_n) \quad (1.1)$$

$$P(X_{n+1} = j|X_n = i) = \frac{w_{ij}}{\sum_k w_{ik}} \quad (1.2)$$

The random walk equations essentially form a Markov chain with the constant teleportation probabilty ρ of jumping back to the cue word "animal".

This model takes into account the two processes indicated by Troyer et al. and creates a more well-defined mathematical framework for further analyses. Intuitively, observed clusters in previous studies would form dense sub-graphs in the semantic network where the random walker will spend some time before jumping to another cluster. To test this claim Abbott et al. used a weighted semantic association network generated by Nelson et al. [10] to simulate IRT for the word sequences of previous studies. They defined IRT as:

$$IRT(k) = \tau(k) - \tau(k-1) + L(X_{\tau(k)}) \quad (1.3)$$

where $\tau(k)$ is the step index where the k_{th} animal first appears. The last term accounts for length of the next word, arguing that longer words take a longer time to remember. For instance, in the following random walk sequence:
$X_1 = "animal", X_2 = "dog", X_3 = "house", X_4 = "dog", X_5 = "cat"$
$IRT("cat") = \tau("cat") - \tau("dog") + L("cat") = 5 - 2 + 3 = 6$
The simulated IRT was shown to have the same characteristics as the experimental results observed by Hills et al. with jumps at cluster switches.

These studies inspire the idea that variations of random walk models can be used to model text documents which are artifacts of the human brain's semantic memory. In the next chapter, we go

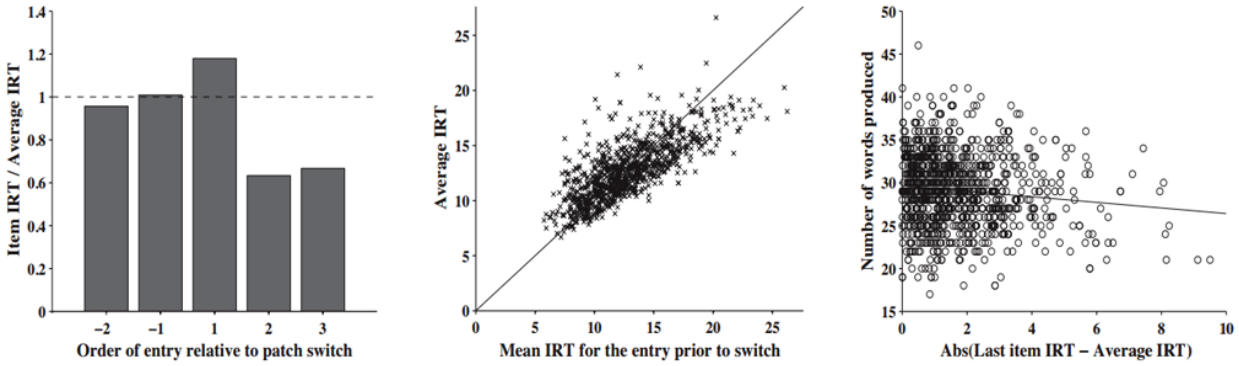


Figure 1.2: Simulated IRT using the random walk model by Abbott et al. [2]

over statistical models of documents and later come back to the notion of random walks and how they can be incorporated in our model.

1.3 Outline

In chapter 2, we give an overview of topic modeling methods for summarizing collections of documents. We then create our associative network of words from a given text corpus, that is to be used by random walk models similar to the memory search process. We will discuss shortcomings of a simple random walk model on this network in modeling dynamics of document writing and its highly contextual nature. We then take help from episodic models of memory to design a model that addresses this contextual behavior. This results in a hierarchical structure representation of the underlying knowledge base.

In chapter 3 we discuss computational methods for extracting the hierarchy described in chapter 2. We start by designing a new random walk scheme whose movements do not lose contextual focus. This modified random walk is used to extract principal core clusters of each context. These cores help us estimate contextually biased associative networks of words and analyzing these context networks leads us to forming a context-term matrix from which the hierarchical structure is built. We will also use this context-term matrix to create summaries of the text corpus comparable to topics of topic modeling methods such as LDA.

Chapter 4 contains evaluations of our summary topics for coverage. We show that our summary

topics include most LDA topics and thus there is no significant loss of information in compressing the corpus into a context-term matrix. We also evaluate our hierarchical structure by inspecting if its layer clusters exhibit high modularity in the associative network.

Finally, in chapter 5 we discuss some sample results of this method and how they reveal additional insights from a document set. Chapter 6 will give a brief review of the tools used and time complexity analysis of our methodology to ensure scalability of our computational framework.

CHAPTER 2

Modeling Document Generation and Underlying Knowledge Structure

2.1 Problem Statement

Our goal is to devise a model for document generation and train it using large sets of documents. We then want to use this model to create a representation for abstract contexts and extract the contexts available in the document set. To this end, we would need to determine if understanding contexts requires us to go deep into sentence structures and use methods like Part-of-Speech (POS) tagging, or if the Bag-of-Words model [9] will be sufficient. We will also determine if we can develop a network scheme for simplifying contextual information in the text corpus.

2.2 Related Work

2.2.1 Topic Modeling

Topic modeling is one of the areas of study that tries to find patterns of words in a text corpus and model statistical behavior of documents. Topic models are usually generative models that try to describe a corpus of text. There has been significant work in this area starting with simple models that have evolved over time. We will briefly review some of these models and their evolution in the following sections.

2.2.2 LSA (Latent Semantic Analysis)

LSA creates a lower-dimensional representation of document collections by decomposing the term-document matrix [12]. If we form this matrix X such that entry X_{ij} gives occurrence of term i in document j, then using Singuler Value Decomposition (SVD), we can create a low-rank approximation to this matrix as $X^{V\times M} = U^{V\times k}\Sigma^{k\times k}V^{Tk\times M}$ such that $k \ll V, M$. In this representation, row i of U provides values that build a representation of word i on the k-dimensional concept space, and column j of V^T gives values that build a representation of document j on the k-dimensional feature space. Therefore, LSA can help do tasks such as conceptual comparison between words or between documents, or finding cluster of similar words.

2.2.3 Mixture of Unigrams

The Mixture-of-Unigrams models assumes there is only one topic for each document and that words in the document are randomly generated from a distribution over words specific to the topic [13]. It also assumes words are selected independently from each other. In this paradigm, a topic is defined as a multinomial distribution over the known vocabulary. The mixture of unigrams model is possibly the simplest model attempting to create simplification of word distributions in a large number of documents into a limited number of distributions regarded as topics. The limiting factor in this model is the assumption that documents only have one topic which in general is not true and over-simplifies the documents.

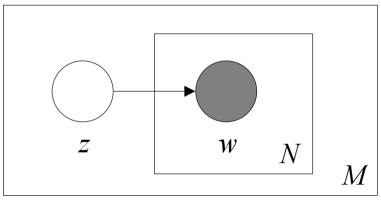


Figure 2.1: Mixture of Unigrams model. Diagram from [3].

$$p(\mathbf{w}) = \sum_{k} p(z) \prod_{n+1}^{N} p(w_n|z) \tag{2.1}$$

2.2.4 pLSI (Probabilitstic Latent Semantic Indexing)

The pLSI model attempts to fix the lack of support for multiple topics in a document in the Mixture of Unigrams model [14]. A document is assigned a unique distribution d over topics. For each word, a topic is randomly picked according to this distribution and then the word is picked randomly from topic's dictionary. The inherent shortcoming of this model is that the document's topic distribution can only be estimated on documents in the training set and there is no way to assign a distribution over topics for a new document. In this sense, pLSI does not present a complete generative model.

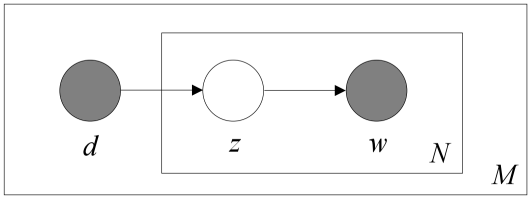


Figure 2.2: pLSI (Probabilistic Latent Semantic Indexing) model. Diagram from [3].

$$p(d, w_n) = p(d) \sum_z p(w_n|z)p(z|d) \tag{2.2}$$

2.2.5 LDA (Latent Dirichlet Allocation)

LDA [3] is the most common algorithm currently used for topic modeling often with slight variations. LDA extends the Mixture of Unigrams model in the same way as pLSI by assigning a distribution over topics to a document, with the key difference that this distribution is itself generated from a Dirichlet distribution with parameter α. With this modification, LDA proposes a complete generative model that can assign probabilities to previously unseen documents as well. This generative model tries to statistically describe the observed document set and is essentially a Bayesian framework for fitting a distribution to observed samples of document words.

$$p(\mathbf{w}) = \int_\theta \left(\prod_{n+1}^{N} \sum_{z_n=1}^{N} p(w_n|z_n;\beta)p(z_n|\theta) \right) p(\theta;\alpha)d\theta \tag{2.3}$$

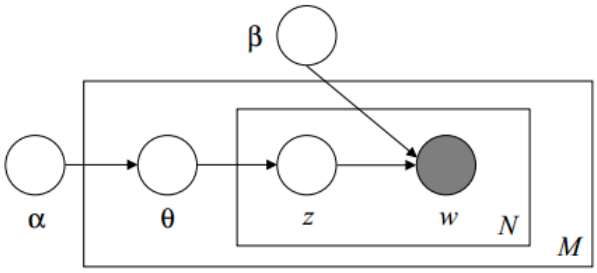


Figure 2.3: LDA (Latent Dirichlet Allocation) model.

Thus after fitting the LDA model to a text corpus, the latent variables would give us parameters of distribution over topics for the corpus, and distribution over words in vocabulary for each topic. These topic dictionaries are stored in vectors that form the β matrix.

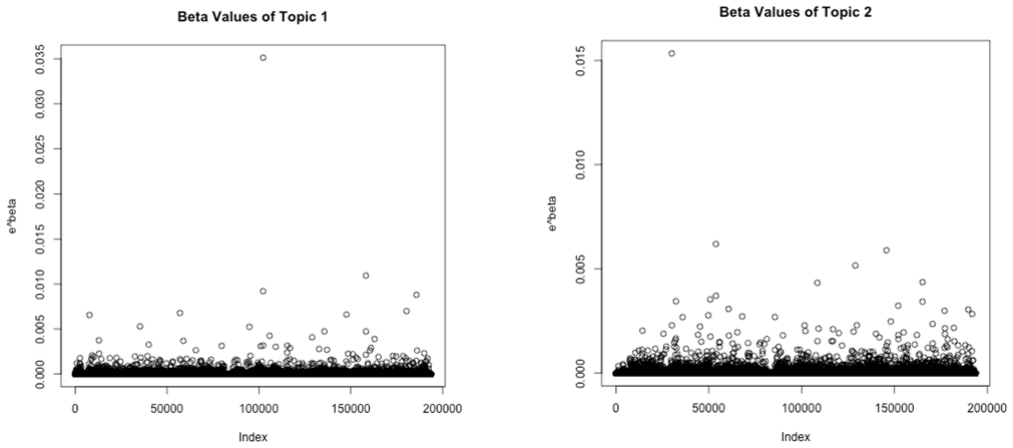


Figure 2.4: LDA topic distribution samples

The LDA model statistically describes observed documents but it does not particularly model the dynamics of writing documents. Another drawback of LDA is the heavy computational cost for fitting the model.

2.3 Network Model

We create a weighted associative network of words from the corpus whose edge weights show strength of ties between words. This network can then be used for random-walk-based models

described in section 1.2. The tie between two words in the network is weighted by the number of documents they both appear in. This will give an empirical estimate of the relative step probabilities in the random walk when going from one node to one of its neighbors.

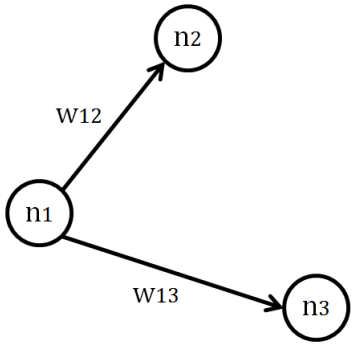


Figure 2.5: 3 nodes in the network with respective edge weights

$$\frac{w_{12}}{w_{13}} \approx \frac{N_d(n_1 \cap n_2)}{N_d(n_1 \cap n_3)} \tag{2.4}$$

An appropriately designed random walker which behaves similar to a document generator should spend considerable time in areas of the network related to only a few contexts. These contexts cause dense clusters of words to form in the associative network. In order to understand how contexts control the random walk process and to find a clear definition of contexts, we need to first analyze the structure of the network. We will then design a respective random walk scheme that corresponds to contextually coherent movements of a document generator.

2.4 Network Structure

To better understand the network structure, we first examine distribution of popularity among words in the network. We need a centrality measure for calculating how frequently the random walker passes a word, for which we use the Google PageRank algorithm [11]. PageRank on co-occurrence network is shown to reveal important words of a text corpus better than simple measures like word frequency [15]. It is also found to provide prominent words in human memory when applied to semantic networks [7].

2.4.1 Google PageRank

PageRank [11] calculates visit probability of a random walker for every node on the network. For a random walker that at each step either jumps to one of the neighbors according to their ingoing edge weight, or teleports to a random node on the network, the steady-state visit probabilities are solutions to the equation:

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in M(i)} \frac{A_{ij} PR(j)}{O(j)} \tag{2.5}$$

Where $PR(i)$ is the PageRank of node i, A is the network incidence matrix, $M(i)$ is the set of outgoing neighbors of noide i, $O(j) = \sum_k A_{jk}$ is the outdegree of node j, N is the number of nodes in the network, and $1 - d$ is the teleportation probability. Defining matrix M as

$$M_{ij} = \begin{cases} A_{ij}/O(j), & \text{if } O_{ij} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

and naming the PageRank vector r the equation can be written as:

$$r = dMr + \frac{1-d}{N}\mathbf{1} \tag{2.7}$$

Then using $Er = \mathbf{1}$ where E is the all-ones matrix, we have

$$r = \left(dM + \frac{1-d}{N} E \right) r \tag{2.8}$$

which is a problem of finding the principal eigenvector as M is by construction a stochastic matrix. A simple method to find the PageRank vector is using the power method [48].

2.4.2 Word Generality and Network Structure

Examining the PageRank distribution of words in the associative network leads us to measuring relative generality of words in the dataset. That is, are words relatively similar in their generality or is there drastic variation among them. The PageRank distribution for two of our datasets are shown in figure 2.6.

The CCDF (Complementary CDF) of PageRank values in both networks shows near-power-law distribution and reveals presence of central nodes with very high PageRank that disturb a random walker's localization.

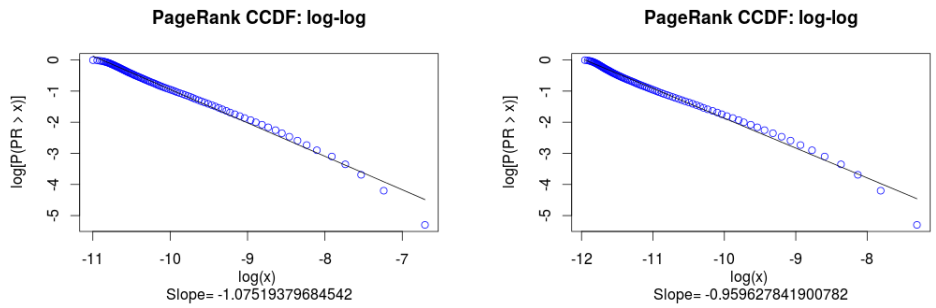


Figure 2.6: Complementary CDF in log scales for the NSF (left) and Cafemom (right) datasets.

Such scale-free networks typically show hierarchical modularity as put forth by Ravasz et al. [16] who analyzed a synonymy network extracted from the Merriam Webster dictionary. There are central cores in the network, words such as "research", "grant", and "study" in the NSF abstracts dataset, that are connected to most other nodes in the network. Moreover, dense clusters in the network can also have different levels of conceptual granularity. For instance, we may see General Physics as a concept, or Quantum Mechanics in particular, and in even further detail smaller concepts such as Hadron Colliders or Semiconductors. The more general concepts are dense clusters that are more *central* in the hierarchically modular network, or dense clusters whose words have higher PageRank values.

The hierarchical structure of the network will have implications on how we devise the random walk model. The simple random walk at this stage describes memory recollection and not document generation. Therefore, "general" words that are strongly connected to most nodes of the network create very frequent random jumps between clusters. This prevents the random walk from staying in contexts and makes it switch to many contexts arbitrarily, which is not how documents are written. Therefore, if we simply use common community finding algorithms, the ouctome solution will be the undivided network showing up as one large community and clusters are not distinguished. We would need aggressive filtering of main nodes or their incident edges to reveal these structures. This would bring in ad-hoc filtering based on keyword detection and edge significance measures to fabricate the network. To avoid such practices we need to tailor our random walk model to the process of writing documents that refrains from straying out of context.

For a better understanding of document generation and contextual coherence of documents, in the following section we continue by pointing out differences between document generation and the memory recollection task, and also clarify our definition of contexts.

2.5 Modeling Document Generation

While the memory recollection experiments and models provide an insight on how the brain accesses stored information, they do not specifically describe the document generation process. A few key differences between these two tasks include:

(i) In memory recollection, there are erratic jumps to other clusters, while document generation is centered around certain clusters that relate to a *"context"*.

(ii) The memory recollection experiment is cued by a single core word, but in document generation the random walk has a central "area" rather than one core word.

(iii) Another difference between memory recollection and document generation is regarding the presence of general words, which may be limited in the first, but very frequent in the latter.

These differences make the regular random walk model insufficient to model document generation and its heavily contextual nature.

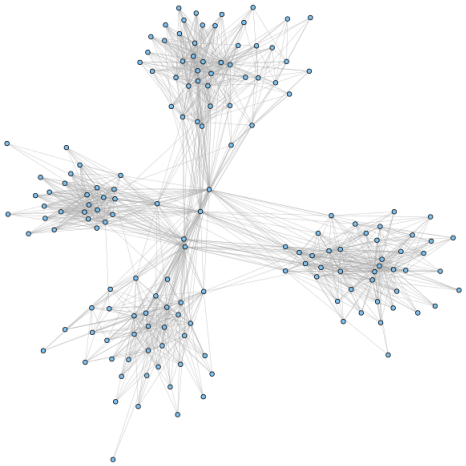


Figure 2.7: A small sub-graph of the network showing a few general nodes that connect contexts

2.5.1 Contextual Memory

A class of memory models that address challenges presented by contextual behavior of memory access are retrieval-based models that put emphasis on memory's episodic nature [17]. These models propose that episodic observations are, without compression and encoding, stored as traces in memory. At the time of retrieval from memory, one makes a cue to these episodic traces and retrieves traces that are more related to the current context of retrieval whose information is contained in the cue.

One of the most famous examples of retrieval-based models is *Minerva 2* by Hintzman [18] [17]. This model essentially views each trace as a vector of values over a set of features. An episodic experience creates one such trace and all traces collectively form a matrix T whose rows correspond to these traces, so T_{ij} is the value of trace i on feature j. At the time of retrieval, the context of retrieval imposes a similarly made feature vector and this probe vector p is used to collect the closest traces from memory by a simple dot product. Therefore, the strength of inducing trace i by probe p is given by:

$$s_i = \sum_{j=1}^{N} p_i T_{ij} \tag{2.9}$$

Where N is the total number of features. The echo vector c returned from memory is then aggregated from traces based on their activation level. In the original implementation of Minerva 2, an activation vector a with $a_i = s_i^3$ is used to invoke traces, instead of directly using s as activation signal. This boosts strong signals and reduces noise, but does not imply any theoretical changes.:

$$c_j = \sum_{i=1}^{M} a_i T_{ij} \tag{2.10}$$

M here is the total number of traces in memory.

A working example of this system given again in [18] sheds light on how this approach can model memory's behaviors such as associative recall: Imagine the traces stored in memory contain faces and their associated names. In this case one can set aside 10 features for faces indexed $j = 1$–10 and 10 features for names indexed $j = 11$–20. When a face is seen, the first 10 entries $p_{j=1\text{–}10}$ of the probe vector are filled with the face feature values and episodic memory traces are

invoked. The content of the echo signal in second 10 entries $c_{j=11\text{–}20}$ can be examined for the associated name.

An application of this memory model in modeling text and document writing has been devised by Kwantes [19] [17]. The key idea is that each document provides one context where a combination of words is seen. Therefore, a row of the term-document matrix whose entry (i, j) is equal to the frequency of word i appears in document j, can be regarded as the memory trace of word i. In this setting, by providing a vector p that is a cue word's distribution across contexts, one can probe the memory traces stored for words and retrieve those similar to the cue word.

To bring context into the picture, in addition to the probe word, Kwantes added a second word to the equation to "prime" memory traces. For instance, if the word "bank" with two meanings related to "river" and "finance" is to be probed, a prime word such as "boat" can be added with the following scheme to disambiguate the meaning of "bank". A new echo vector is generated by:

$$c_j = \sum_{i=1}^{M} a_i^{\text{probe}} a_i^{\text{prime}} T_{ij} \tag{2.11}$$

The prime word's activation vector creates a contextual representation of each trace i, by calculating $a_i^{\text{prime}} T_{i\star}$, which is then used by the probe vector's activation signal as before. The contextually adjusted traces in the example above would already have a bias towards "boat"-related words and after being probed by "bank" they will return contextually relevant words like "river".

We base our contextual semantic memory model on the episodic memory model in Minerva 2, and borrow ideas from the Kwantes model but build a different setup of memory traces and contextual retrieval. We regard each document as a memory episode that has a trace in memory as a distribution over words. A context can be defined by a group of very similar documents that share detailed topic of discussion so their traces are very similar. The actual traces stored in memory can therefore consist of context traces instead of individual documents. In this model, each row of our memory trace matrix T is a context's trace, so entry T_{ij} is context i's assigned value to word j. It is worth noting that in contrast to the Kwantes model, in our semantic model words are the features and contexts are the stored traces described in Minerva 2.

To model contextual retrieval we make use of the structure in our hierarchically clustered co-

occurrence network. Our conjecture is that clusters at the bottom level of the hierarchy (dense clusters at the lowest PageRank level) each correspond to a group of very similar documents. A cluster at the bottom layer consists of very detailed words of a very specific concept. The documents that contain this group of words have to be very similar in topic and therefore the higher-layer words and clusters that these documents incorporate should also be similar. This claim is partially tested in chapter 4.

The idea that bottom-layer clusters correspond to a group of similar documents helps us build contexts —which we also think of as a group of similar documents— around these clusters. A context made up of the group of similar documents corresponding to a bottom-layer cluster, can be identified by words of this cluster. In other words, this cluster contains the set of unique words for its related context and therefore having only this group of words as a "seed set" clarifies which context is under discussion. We call this set C containing the bottom-layer cluster words, the *"context core cluster"*.

To retrieve a context's respective feature vector of words, the memory trace matrix is probed with a probe vector p that contains high values in entries $p_j : j \in C$ and zeros for other words. The returned echo vector can then be examined to find its high values which point to other contextually important words.

This representation of semantic memory can further be simplified by extracting clusters of contextually correlated word groups in higher PageRank layers. We will see layers as PageRank intervals and a layer's words will be words with PageRank value inside the interval. For a layer in a range of relatively high PageRank values, if L is the set of words in this layer, the matrix representation of contexts stores n_c (number of contexts) vectors of length $|L|$. The hierarchical modularity of our co-occurrence network implies that words of such a high PageRank layer rise mostly as a group and there are certain grouped combinations of layer words that show up in many contexts. By extracting these underlying word groups at each layer, we will be able to create a simpler hierarchical representation of contextual memory that in each layer consists of the layer clusters of words. Then a context's trace over words can alternatively be described by a trace over layer clusters at different layers.

Finally, we have a hierarchical structure of concepts at different layers —provided by layer clusters— and how these concepts relate across the layers. In this representation, a layer cluster is viewed as a concept node in the respective layer, and edges between these nodes at different layers show their correlation in contexts. Each context's trace on this hierarchical structure will be in the form of highlighted nodes and paths. A small example piece of one such structure is shown in figure 2.8.

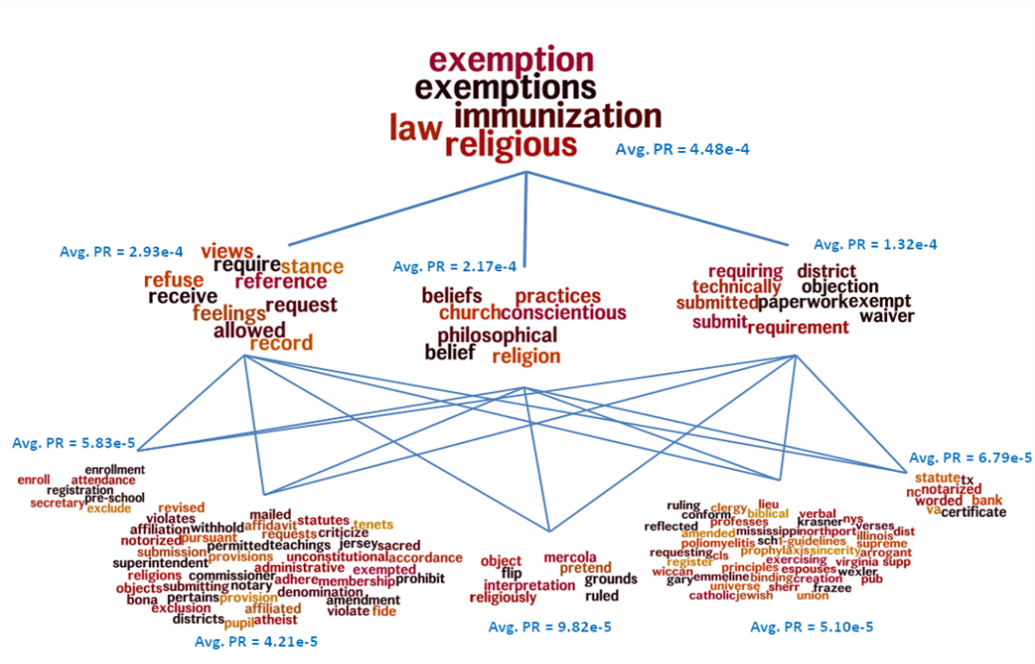


Figure 2.8: A sample piece of a hierarchical structure for one of our datasets containing forums of parents discussing vaccination on a website named Mothering.com. This piece is about "exemption from vaccination" for children. These concepts arise from parents' determination to not vaccinate their children, and searching for strategies to avoid required vaccination, such as religious exemption laws. Average PageRank of nodes in each layer cluster words is noted on side.

CHAPTER 3

Methodology

Our strategy for extracting the hierarchical structure described in section 2.5 can be realized in the following steps, drawn in figure 3.1.

1. Finding contexts by extracting core context clusters at the bottom PageRank layer of the co-occurrence network. We devise a modified random walk scheme to trap the walker in these detailed context clusters, then use random-walk-based community detection tools to find these clusters.

2. Finding *"context profiles"*; each context's feature vector that provides significance of any word in the context. Context profiles are calculated by creating a contextually biased version of the co-occurrence network and comparing PageRank of words in the new network to their prior PageRank in the original co-occurrence network. These context profile vectors together form the context-term matrix.

3. Extracting clusters at different layers of the PageRank spectrum. We divide the PageRank spectrum in overlapping windows (layers) and get the corresponding block of the context-term matrix for words in this layer. Matrix factorization tools are then used to find recurring clusters of words in the layer.

4. Building the hierarchical structure from layer clusters. Connections between these layer clusters can be drawn by either inspecting cluster overlaps from one layer to next, or by examining contextual co-occurrence of layer clusters and finding groups of clusters that tend to come together in contexts. Note that layers are overlapping windows and clusters in two consecutive layers can share words in the overlapping part of these two windows.

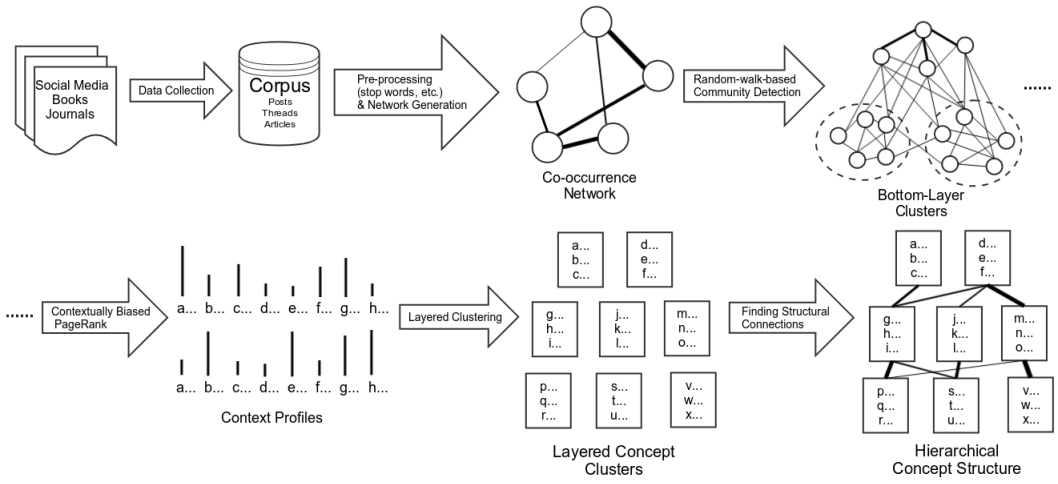


Figure 3.1: Breakdown of steps to extract the hierarchical concept structure.

3.1 Random Walk with Reflection

To account for differences indicated in document generation from the memory recollection task, we introduce a *reflection* probability function in the random walk process. This reflection scheme is mainly driven by the notion that documents are consistent in bringing up lower-layer context core clusters and the appearance of general words in the document does not make them stray to another context. Therefore, when a random walker reaches a high-PageRank general word's node, it should most likely reflect back to the low-PageRank cluster where it came from. This reflection scheme will keep the random walker in focused clusters of detailed words at the bottom PageRank layer of the network and help us extract these cores.

A regular random walker has a jump probability to one of the neighbors of its current node proportional to their corresponding link weight. The probability of going to a neighbor j of node i at time step $t+1$ if the walker is in node i at time t is:

$$p(s_{t+1} = j | s_t = i) = \frac{w_{ij}}{\sum_k w_{ik}} \tag{3.1}$$

The reflection probability introduced after a step from i to j is:

$$p_r(i \to j) = \sigma\Big(c\big(PR(j) - PR(i)\big) - \big(PR_0 - h.PR(j)\big)\Big) \tag{3.2}$$

where $\sigma(t)$ is the sigmoid function

$$\sigma(t) = \frac{1}{1 + e^{-t}} \tag{3.3}$$

and c,h, and PR_0 are scaling and shifting constants chosen according to the corpus PageRank distribution. Changing the shape of this reflection function can control granularity of contexts for the random walker, which given a certain function may only reflect from the very general words, or it can alternately stay in more granular contexts by using a function that also reflects from medium-PageRank nodes.

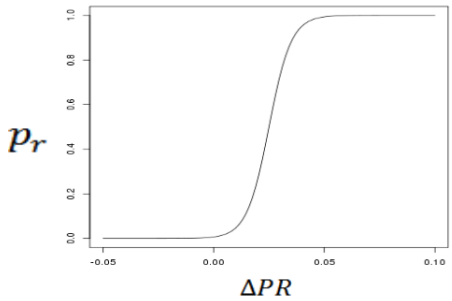


Figure 3.2: Random walk reflection probability as a function of PageRank difference. The shift in the sigmoid function is also a linear function of the destination PageRank.

The incorporation of the reflection function affects visit probabilities of nodes by the random walker. The effective PageRank with such random walks for two of the datasets are shown in 3.3.

The modified PageRank distributions have a curved tail and high-PageRank nodes are suppressed.

3.2 Hierarchical Clustering

Incorporating the reflection function helps the random walker focus on the bottom layer of the network and get trapped in highly specific concept clusters. These are clusters of words that appear in a small group of documents that are very similar in topic. These word clusters can thus represent contexts available in our dataset. With a given parameter set of the reflection function, one can use any random-walk-based community detection algorithm to find these bottom layer clusters. In this work we use the InfoMap algorithm to find our granular word communities. Further discussion on

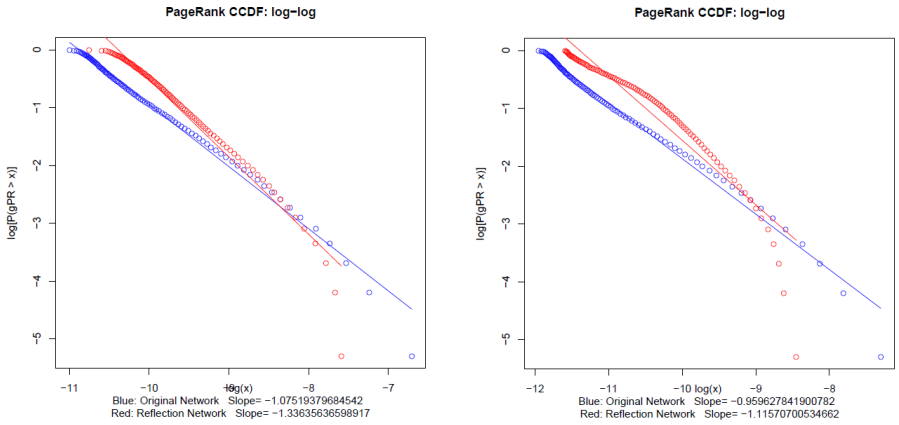


Figure 3.3: Change in distribution of effective PageRank with modified random walk for the NSF (left) and Cafemom(right) datasets

community detection tools and InfoMap can be found in chapter 6. Table 3.1 shows examples of bottom layer clusters extracted with one set of reflection parameters.

The network's cluster structure tends to be continuous on the PageRank spectrum and these bottom clusters do not necessarily occur in similar PageRank values. Instead, the average PageRank of cluster nodes for the set of bottom layer clusters falls in a range of values that is not easy to retrieve from the network without prior knowledge of cluster compositions. The fact that average PageRank of the clusters are not exactly similar, along with difference in PageRank value of words inside the cluster, makes their extraction somewhat sensitive to changes in the reflection parameters. While many clusters appear in different combinations of the parameters that provide a reflection function with its transient phase located somewhere in the middle of the PageRank spectrum of the dataset, in some specific combinations some of the clusters do not appear. In most cases the same cluster shows up slightly different across extractions that use different parameter combinations. This dissimilarity can be a few different words between the two extractions or sometimes some dropped words. These differences are due to adjustments made by the reflection function as well as different convergence points of the non-convex community detection problem. Table 3.2 shows 4 sample observations of the same cluster extracted with different parameter sets of the reflection function.

	Bottom-Layer Cluster Words
Cluster 1	redshift dwarf quasars faint luminosity photometric photometry extragalactic redshifts milky dwarfs globular halo hubble bang nucleosynthesis luminous nebulae magellanic keck spectrograph circumstellar near-infrared binaries spiral primordial envelopes lensing cool sight colors elliptical mid-infrared interiors atmospheres big radial
Cluster 2	semiparametric bootstrap econometrics resampling estimator covariates censored nonparametric estimators smoothing inferential asymptotically bayes bayesian covariance econometric conditional regression markov distributional doubly inference multivariate inferences parametric fitting likelihood restrictions confidence
Cluster 3	csems academically financially scholarships low-income csem tutoring bachelor baccalaureate counseling advising graduation disadvantaged majoring scholarship enroll talented forty internships internship enrollment placement succeed
Cluster 4	pituitary estrogen gonadal hypothalamus neuroendocrine neuropeptide gland testosterone neuroanatomical steroids neuropeptides neurochemical anterior adrenal progesterone gnrh glands androgen ovarian endocrinology steroid releasing ovary hormones hormone secrete pregnancy prolactin endocrine forebrain thyroid vasopressin hormonal circulating dopamine posterior secretion aplysia transmitters ganglia secreted fetal exert
Cluster 5	macroeconomic macroeconomics monetary asset inflation assets fiscal volatility portfolio trading pricing nominal microeconomic prices shocks returns money goods economies economists stock finance consumer price derivative

Table 3.1: Five examples of bottom-layer clusters extracted from a dataset of NSF funding abstracts. These clusters are related to relatively detailed contexts from "astronomy", "statistics", "education", "biology", "economics".

	Bottom-Layer Cluster Words
1st Observation	redshift dwarf quasars faint luminosity photometric photometry extragalactic redshifts milky dwarfs globular halo hubble bang nucleosynthesis luminous nebulae magellanic keck spectrograph circumstellar near-infrared binaries spiral primordial envelopes lensing cool sight colors elliptical mid-infrared interiors atmospheres big radial
2nd Observation	extragalactic ast galactic galaxy galaxies redshift telescopes dwarf stellar cosmology star telescope quasars cosmological photometry interstellar faint luminosity photometric milky astronomers astronomical sky halo stars globular redshifts astrophysical astrophysics bang dark hubble gravitational dwarfs astronomy brightness universe binaries magellanic nucleosynthesis spiral luminous ccd observatories keck observatory observational sun cool binary ionized primordial holes bright disk abundances distant observing colors masses big camera massive nearby sight elliptical schmidt radial clues
3rd Observation	extragalactic dwarf galaxy redshift supernovae ast galactic photometry luminosity quasars faint galaxies astronomers supernova photometric stellar milky cosmological interstellar telescopes telescope sky redshifts stars star cosmology hubble halo globular dwarfs brightness astronomical astrophysical bang nucleosynthesis luminous nebulae magellanic circumstellar spiral observatories binaries astrophysics keck dark spectrograph explosions cool gravitational remnants planets bright ionized atmospheres near-infrared lensing ccd primordial elliptical angular colors envelopes giant remnant interiors sight radial schmidt
4th Observation	redshift faint milky redshifts quasars photometry halo luminosity photometric globular hubble magellanic dwarfs dwarf binaries luminous extragalactic keck lensing spiral colors schmidt sight brightness bright radial

Table 3.2: Four different observations of the same bottom-layer clusters extracted from the NSF funding abstracts dataset. These clusters are observed in community detection outcome for four different parameter settings.

To reduce sensitivity to parameter settings, we run a sweep on different parameters of the reflection function and with each parameter set get the clusters extracted by the community detection algorithm. The results of these different extractions need to be reconciled to create one set of clusters. Multiple observations of the same cluster across different runs are merged together to form one inclusive version of the cluster. These observations of the same cluster are detected by finding clusters across different runs with high Jaccard index $\frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$. After similar clusters between different community detection results are merged, we take larger clusters, with 5 or more words, to include in our set of contexts. The clusters of table 3.2, along with other observations of the same clusters, are thus merged to result in the following:

```
extragalactic ast galaxy dwarf galactic redshift supernovae galaxies photometry
quasars faint stellar luminosity astronomers telescope interstellar photometric
telescopes supernova milky star cosmological cosmology stars sky redshifts hubble
halo astronomical globular dwarfs astrophysical brightness bang astrophysics
nucleosynthesis circumstellar luminous nebulae dark binaries astronomy observatories
gravitational magellanic keck spiral spectrograph planets universe cool explosions
remnants ccd ionized atmospheres dust near-infrared observatory bright lensing
observational primordial angular sun elliptical abundances planetary distant colors
wavelengths envelopes binary disk holes giant observing camera massive remnant sight
nearby big radial schmidt night illumination
```

3.3 Context Networks

The clusters extracted from the lowest layer are about a very detailed concept that do not occur often throughout the dataset and almost always occur together as a group. The small group of documents that contain one of these word clusters are about the very particular concept that the word group defines. Such similarity helps us regard the document group defined by this word cluster as a "context" described in section 2.5.1.

The importance of words, and their association strength, is different from one context to another. In the global co-occurrence network, the co-occurrence recorded between two words in the higher PageRank layers is the sum of the pair's co-occurrence in all different contexts. In a given context, the word pair's co-occurrence would be only a fraction of this value. This information is lost in the single co-occurrence value calculated for the pair from the entire dataset.

Ideally, one could create a context's co-occurrence network by finding the co-occurrence network generated from the set of documents that define the context. Now instead of the document set and their complete wording, we have the word cluster containing these documents' distinctive set of words and we want to use this word set and the global co-occurrence network to create a contextually biased association network that is an estimation to the contextual co-occurrence network. The distinctive feature of words in bottom layer clusters, as opposed to higher layer words, that lets us do such estimation is the assumption that these words are mostly "contextually pure" and their global co-occurrence values are caused by only one context. The co-occurrence values between this word set and the rest of the network contain contextual information that helps us create a contextually biased network as described in the following.

First, we estimate the contextual occurrence count for words outside the core context cluster; an estimate for their occurrence count in the context's relative document set. Since the core group of words mostly occur in this document set, the co-occurrence between pairs inside the group and also the co-occurrence between one of these words and a word outside the cluster comes mostly from the same document set. For a given word outside this core group, its co-occurrence with a word inside is indicative of how many times the outside word appeared in documents where the inside word was present. To estimate the number of times the outside word has appeared in documents where most of this core are present, we take the median of co-occurrence values between the outside word and all words inside the core cluster. In other words, for a set of core words C and a node i outside of this set, as shown in 3.4, we estimate contextual occurrence count N_i^C as:

$$N_i^C = median(w_{ik}, \forall k \in C) \tag{3.4}$$

If two words (i, k) are not connected, we put $w_{ik} = 0$ when calculating the median.

Once we have the contextual occurrence estimate for every word in our vocabulary, the next step is to estimate contextual association between pairs of words. As mentioned above, we assume the occurrence count and co-occurrence of words inside the core cluster and their co-occurrence values with any other word are contextually pure. This assumption means that for word pairs

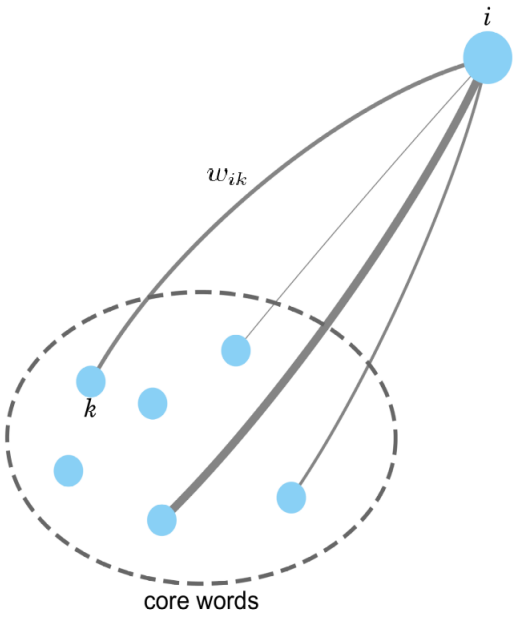


Figure 3.4: For a bottom-layer cluster of words and a given outside word, the co-occurrence values between the outside word and words of the cluster carry information about how many times the outside word appears in the document set containing the cluster words.

(i, k) where $k \in C$, the contextual co-occurrence is similar to their global co-occurrence in the corpus. For a word pair (i, j) outside the core cluster, we need an estimator function for contextual association. Two simple examples of such estimator functions are:

$$f(i,j) = \frac{N_i^C + N_j^C}{N_i + N_j} w_{ij} \tag{3.5}$$

$$g(i,j) = \frac{min(N_i^C, N_j^C)}{min(N_i, N_j)} w_{ij} \tag{3.6}$$

Where N_i^C, N_j^C are contextual occurrence estimates of (i, j), and N_i, N_j are their respective global occurrence counts in the corpus. The "min" function is more conservative in assigning high association values and performs better around zero, while the "sum" function provides more associative boost at the risk of large errors for low values. We use the "sum" function $f(i, j)$ to exploit this boosting behavior which makes it highlight contextually significant words clearer.

Now with contextual associative weights calculated, the context network is complete. It is

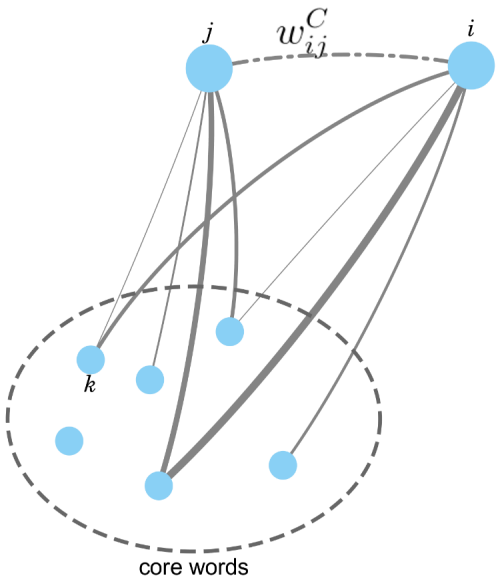


Figure 3.5: For a pair of words outside the core context cluster C we estimate their contextual association w_{ij}^{C} using the estimated contextual occurrence counts N_i^C, N_j^C

worth noting at this point that in a special case, the assumption that co-occurrence value between a word inside the bottom layer core cluster and an outside word is all coming from the same context does not hold true. That is at times when a word k' belonging to two contexts C_1 and C_2, or sometimes a vague word with multiple meanings, appear in the bottom layer cluster. Because of the hard clustering approach, these dual-context words get assigned to one of their relative context clusters at the bottom layer. Let's assume this k' is assigned to the context cluster of C_1. Now if we are estimating co-occurrence values for context C_1, for a word i' in the higher layer that relates to the other context C_2 but not C_1, the co-occurrence value $w_{i'k'}$ is high but $w_{i'k'}^{C_1}$ should be low. The high co-occurrence value between k' and words related to context C_2 can make them appear slightly significant in the context network of C_1. In such cases, for other words $k'' \in C_1$ that $k'' \neq k'$ the co-occurrence value $w_{i'k''}$ is small.

In order to fix these corner cases, for any outside word, we examine it's co-occurrence values with words inside the context and remove outlier edges among them. In the case described above, for context C_1 the co-occurrence value $w_{i'k'}$ would be an outlier compared to other co-occurrence values $w_{i'k''}$. The outliers are detected with simple chi-squared scores and thresholding. For an

outside node i and cluster words $k \in C$ chi-squared scores χ^2_{ik} are calculated as:

$$\chi^2_{ik} = \frac{(w_{ik} - m_i^C)^2}{\sigma_i^{C^2}} \tag{3.7}$$

Where m_i^C and $\sigma_i^{C^2}$ are sample mean and sample variance across co-occurrence values w_{ik} over $k \in C$. In our context networks, we remove any edge w_{ik}^C if $\chi^2_{ik} > 2$.

3.4 Context Profiles

Each context network is a biased version of the associative network that puts more emphasis on contextual importance of words and their contextual association. In a specific context, we can assign importance to words by calculating their PageRank as in the global co-occurrence network. The absolute value of PageRank would still be high for the original high-PageRank words of the global co-occurrence network. These are more general words that appear very frequently in different contexts. The new PageRank distribution of words, however, has a contextual bias that implies a shift in each word's PageRank when compared to the original PageRank value. The words with higher significance in the current context will get a larger boost in their PageRank. In order to find significant words in the context, we compare the contextual PageRank of words to their initial PageRank by calculating ratio of the two for a word i: $PR^C(i)/PR(i)$. Figure 3.6 shows a plot of the PageRank increase ratio of words for a given context; or what we refer to as the context's *"PageRank profile"*. For each word, its position on the x-axis is the initial global co-occurrence network PageRank value, and the y-axis position is the word's ratio of context network PageRank to initial PageRank.

For words in a specific range of PageRank values, one can measure the boost observed in PageRank to find contextually important words in that range. We divide the PageRank spectrum on the log scale into a number of bins and examine words in each bin to find outliers among them in terms of PageRank ratio. The number of bins is set to 20 for all datasets studied here. Figure 3.7 is an example close-up of the PageRank profile in a few middle-layer bins, bin 10 through 12, for a context in the NSF abstracts dataset. The context is one defined by the core cluster below:

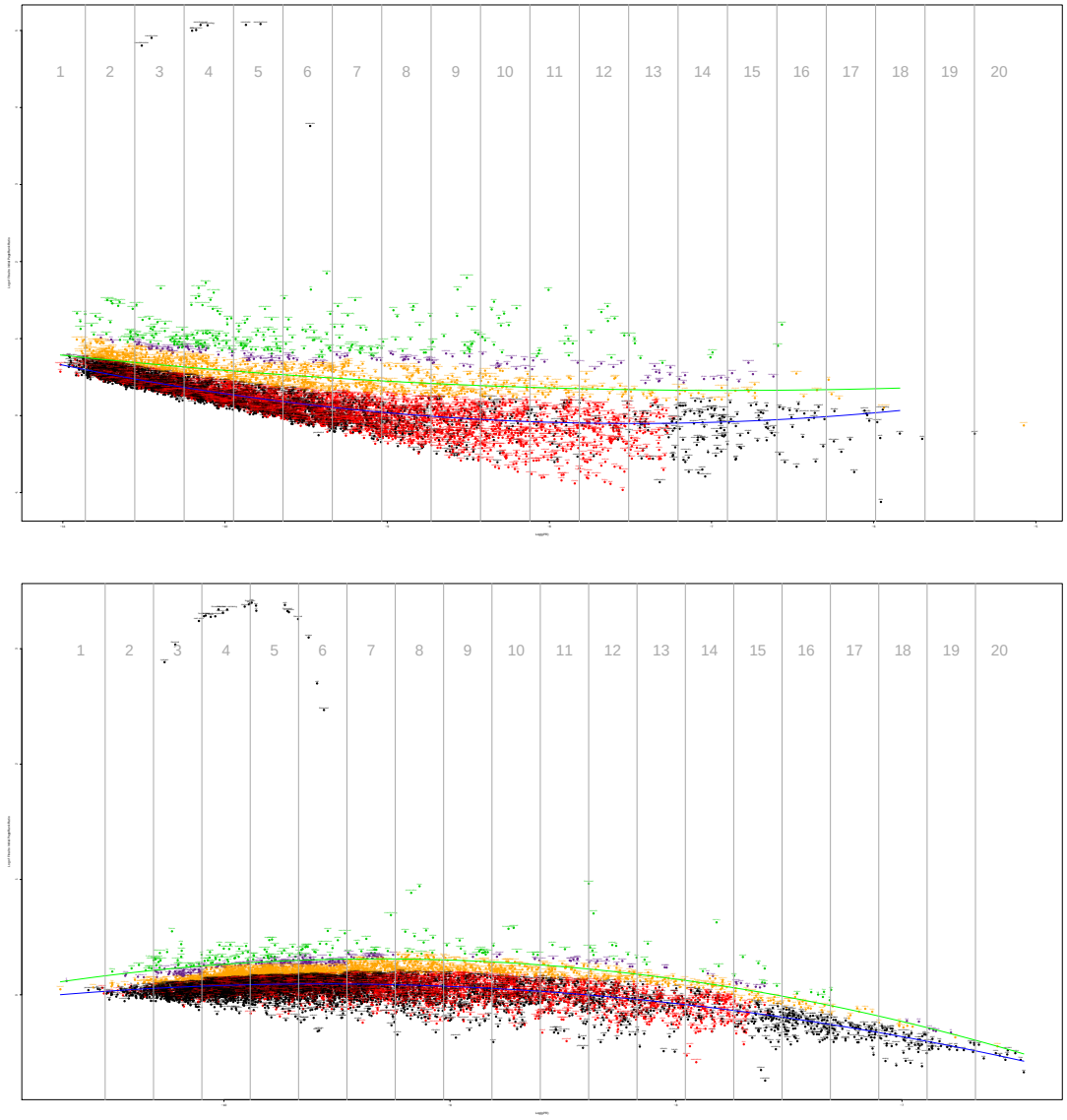


Figure 3.6: Context-to-global PageRank ratio plot (PageRank profile) of a context in NSF abstracts (top) and a context in vaccination discussion forums of Mothering.com (bottom). Each point is a word, with its position on the x-axis equal to the word's global PageRank (PageRank in global co-occurrence Network), and its position on the y-axis equal to the context-to-global PageRank ratio. The core cluster that defines the context gets the highest boost and can be seen in the points on the top left. Each context brings up some words at different PageRank ranges above its core.

```
nanostructured dots nanostructure nanoparticles nanotechnology
nanofabrication nanostructures nanometer nanoscale
```

The core cluster sets the context to a very specific topic about *nanostructures* with core words that in the PageRank spectrum mostly fall into bins 3 to 5. As evident from the PageRank profile, in the middle layers of the PageRank spectrum more general words of the context such as "quantum", "microscopy", "electron", and "metal" come up very strongly. In higher bins 14 through 16, as displayed in figure 3.8, we can see even more general words that are related to the context come up, such as "structures", "surface", "materials", and "properties".

Another example of such patterns can be seen in the profile of the following core cluster in the vaccination discussion forums dataset from Mothering.com:

```
multi-dose thimerosal-containing thimerosal-free mercury-free
mcg formulations syringes thimersol neurodevelopmental epa
formulation vials thimerasol vial micrograms shelves
thimerisol thermisol syringe thimerosol 1990s ban ml
```

The core cluster contains words about vaccine ingredients that the online discussions society mainly blames for the suggested connection of vaccines and autism, like "mercury" and the mercury compound "thimerosal/thiomersal". The cluster itself has lower-PageRank words of the concept, such as "thimerosal-containing" and "mercury-free" as well as different misspelled versions of "thimerosal". In the context's middle-layer bins 10 through 12 shown in figure 3.9 "thimerosal" itself is a very significant word along with "amounts", "contained", "formaldehyde". Among the outliers but a little less significant are words like "neurological", "manufacturers", "fda", and "toxic" which are of other concepts discussed in the context of vaccine ingredients. These are resulting from the ideas circulating in the forums that these ingredients cause neurological disorders, or that manufacturers play a role in the scenario. The higher bins 14 through 16 displayed in figure 3.10 also include "mercury" that is of higher PageRank than "thimerosal" as it is mentioned more in threads. They also show more general words coming up such as "dose" and "safe".

Another important observation in the context PageRank profiles is that in addition to contextually pure core clusters in the bottom layer, the words in higher layers tend to come up as clusters as

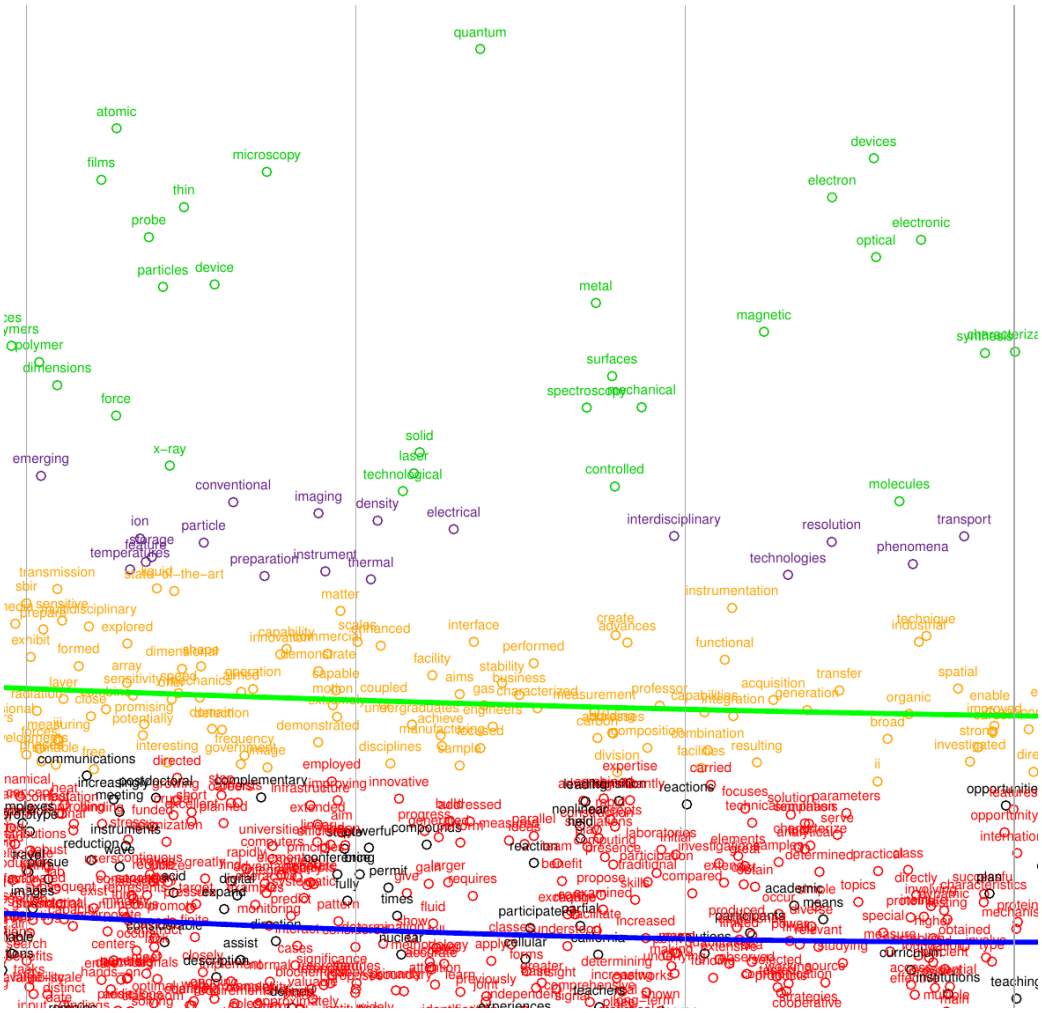


Figure 3.7: Outliers in PageRank profile of a context in NSF abstracts on *"nanostructures"*. The outliers are colored green, purple, and yellow, in decreasing order of significance. This crop shows middle-layer bins 10,11,12 out of the 20 PageRank bins.

well. Figure 3.11 is an example of 3 related contexts and the close-up of each context's PageRank profile in bin 12. Even though each context has its own signature in the higher layers —both in the combination of words it brings up and in their relative significance— one can clearly see a group of words come up in all three: "communication, algorithms, implementation, efficient, power" and some others are either clearly outliers in all three contexts or have relatively high contextual boost values.

Similar patterns can be seen in different PageRank layers for different groups of related context

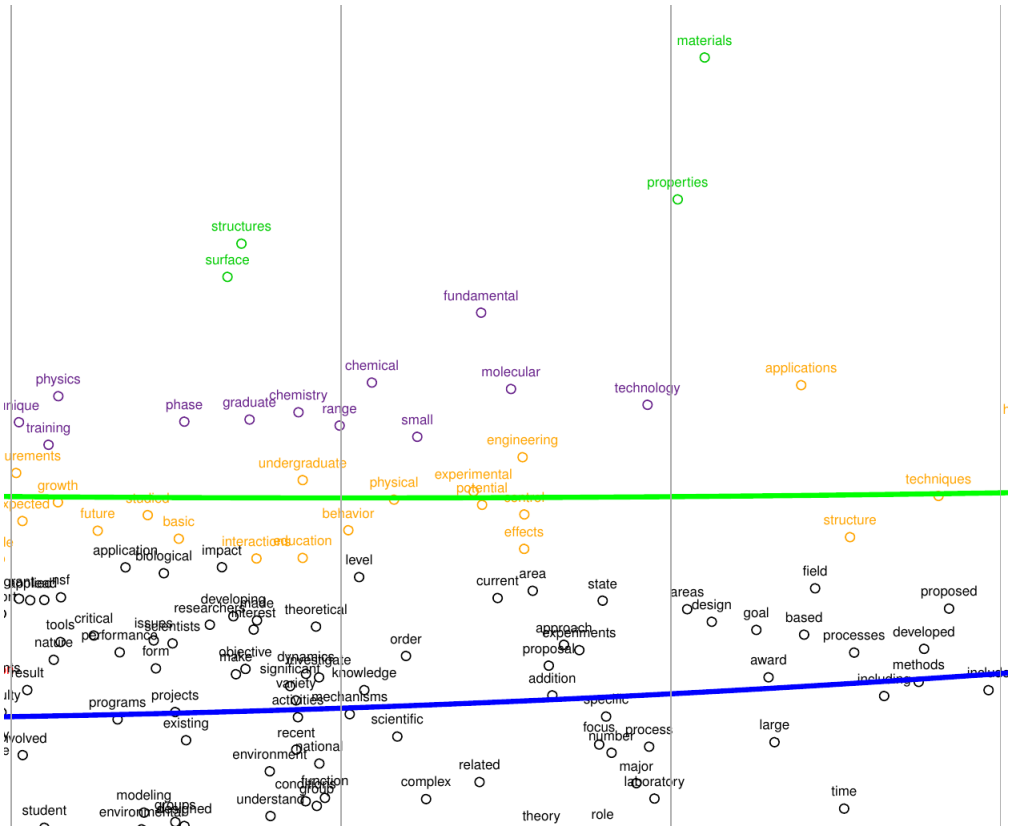


Figure 3.8: Bins 14,15,16 out of the PageRank profile in figure 3.7

and throughout various datasets. Another example is shown in figure 3.12 for three contexts in the forum discussions on Mothering.com in bins 15 and 16. All three contexts are related to medical check-ups and doctor visits and so all of them bring up words such as "doc", "ped", "office", "visit", "practice", and "pediatrician".

Numerous recurring examples of such group co-occurrences at higher layers of the context PageRank profiles support the idea that there are significant groupings of words in the higher layers which are the cause for hierarchical modularity of the co-occurrence network. Each context can touch on multiple clusters of words in a higher layer and bring them up with different intensities, but once a cluster in that layer is hit by the context, most of the cluster words will be brought up. For example, the "medical check-up" contexts mentioned in figure 3.12 talk about two different topics associated with doctor visits. One is the matter of office visits and scheduling the check-ups with a doctor. The other is finding doctors who are understanding of the non-vaccinating parents and are

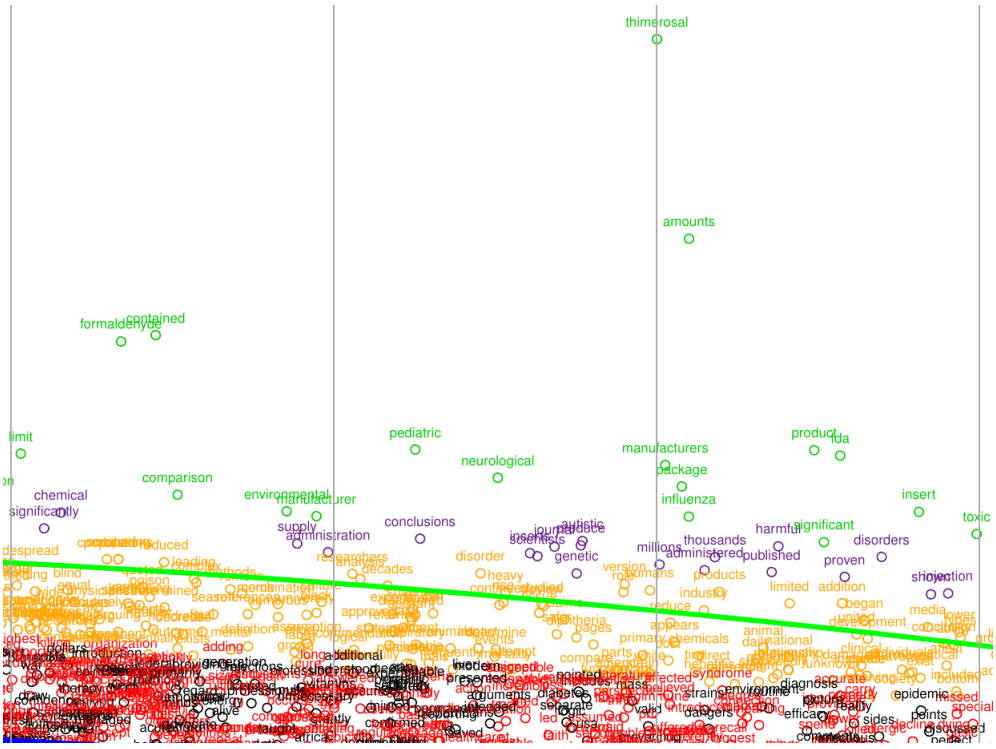


Figure 3.9: PageRank profile of a context in Mothering.com forums discussing vaccination. This crop shows middle-layer bins 10,11,12 out of the 20 PageRank bins for a context about vaccine ingredients.

proponents of homeopathy. Each of these topics has its own clusters of words at different levels of PageRank and each of the three contexts touches clusters of one or the other more strongly; or sometimes only one. Their difference can be viewed as an example in their signature at bin 9 in figure 3.13. While all contexts bring up words related to challenges with doctors themselves through words like "abuse", "naturopath", and "respectful", the first one also has a strong touch on "scheduled", "yearly", "provider" which are only of secondary interest in the third context.

These similarities can also be observed by calculating the overlap at each bin between outliers of pairs of contexts. A simple measure to quickly observe these patterns is the directional overlap of significant outliers shown in green in the plots. If the set of these clear outliers for contexts C_1 and C_2 at bin b are $G_b^{C_1}$ and $G_b^{C_2}$, the directional overlap from C_1 to C_2 can be defined as $\frac{\left|G_b^{C_1} \cap G_b^{C_2}\right|}{\left|G_b^{C_1}\right|}$. A plot of these overlap values for the first and second context of figure 3.12, whose respective index in the dataset contexts are 66 and 30, can be seen in figure 3.14. The numbers shown above each

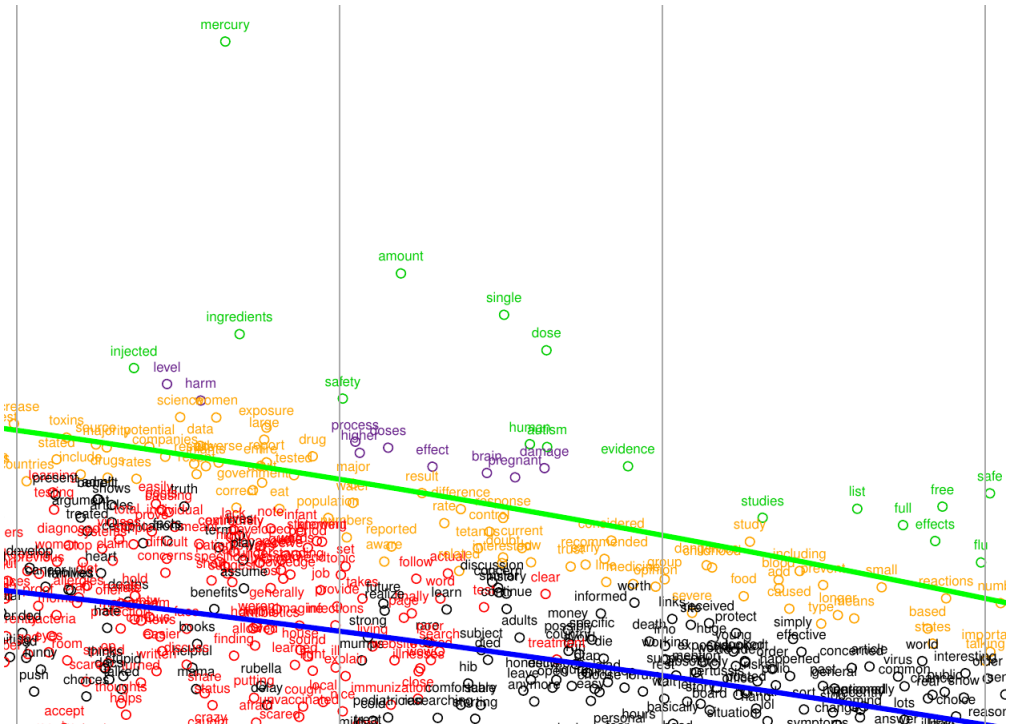
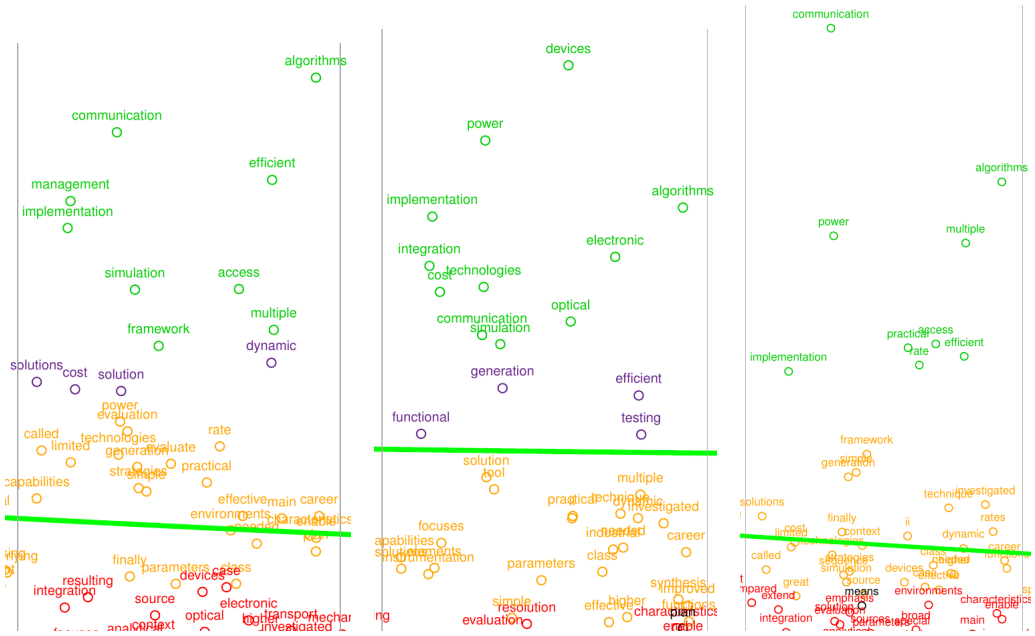


Figure 3.10: Bins 14,15,16 out of the PageRank profile in figure 3.9

overlap point indicate how many of the top ranked outliers of the first context exist in set $G_b^{C_2}$ outliers of the other context. The number below the point indicates how many top outliers in $G_b^{C_1}$ were checked for existence in $G_b^{C^2}$. This number is either 5 or the total number of outliers in $G_b^{(C_1)}$, whichever is smaller. For two similar context like 30 and 66 the overlap looks consistently relatively large throughout the PageRank spectrum. Another case of such similar context pair overlaps can be seen in figure 3.15 between a food-related context and two other contexts about food and homeopathy.

Such grouped movement of words lets us extract concepts, or groups of contextually correlated words, at different levels of PageRank by examining context PageRank profiles. I.e. in different layers we can find significant clusters of words that are correlated in their contextual rise, and then build up a hierarchy of these concepts upward the PageRank layers.



(a) 1st Context (b) 2nd Context (c) 3rd Context

Figure 3.11: Bin 12 of different PageRank profiles for three related contexts in the NSF abstracts dataset. The core cluster of contexts from left to right are:

1) qos congestion end-to-end ip packet tcp routers guarantees packets atm multicast node router admission buffer delays guaranteed routing scalability nodes guarantee destination traffic topologies delay protocol hoc

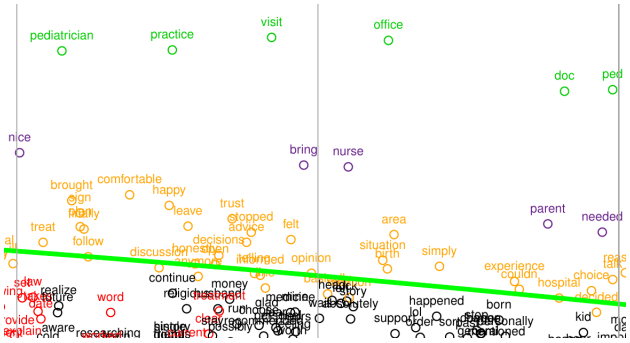
2) cmos interconnect on-chip interconnects ic layout vlsi chips mixed-signal low-power chip ics circuit analog circuits

3) fading multipath decoding cdma equalization multiuser turbo trellis coded wideband dsp reception cancellation shannon isi

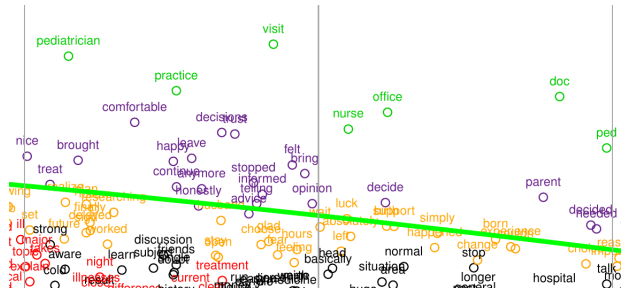
3.5 Concept Hierarchy

3.5.1 Extracting Concept Clusters In Different Layers

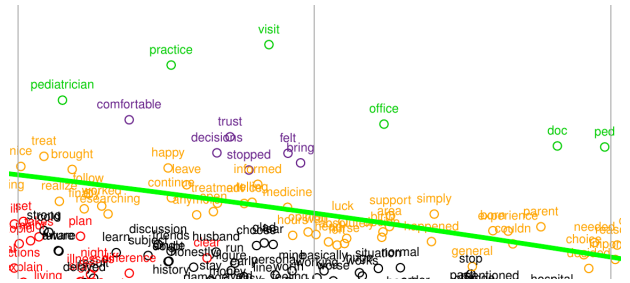
In order to extract concepts at middle layers using context PageRank profiles, one could find the binned outliers in each context, that are contextually significant words, and then create a contextual co-occurrence of words in the middle PageRank layers. In this network the association value between words i, j would be w'_{ij} the number of contexts where the pair appear simultaneously as



(a) 1st Context



(b) 2nd Context



(c) 3rd Context

Figure 3.12: Bins 15 and 16 of different PageRank profiles in the Mothering.com forums datasets for three related contexts about doctor visits. The core clusters of contexts from top to bottom are:

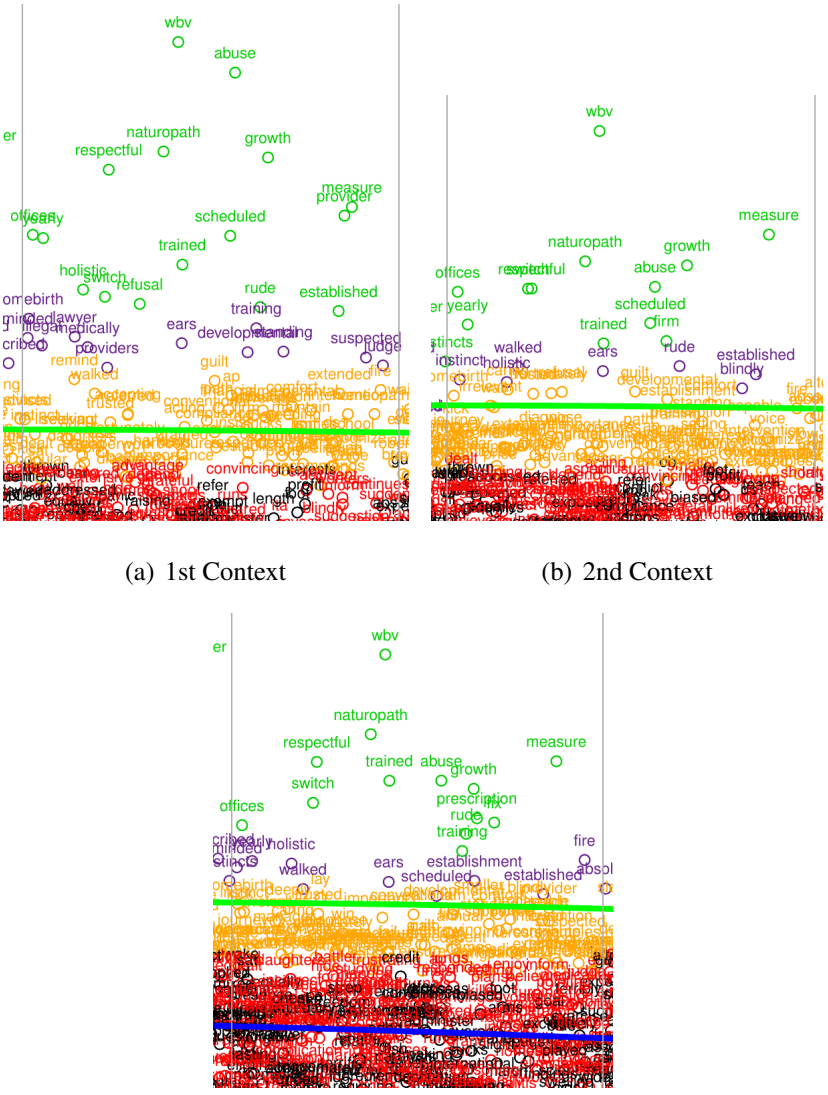
1) wbvs checkups check-ups wbc appts physicals neglectful cya practioner well-child receptionist harassed neglecting hassled ins referrals referral hmo pedis harass emergencies fp harassment

2) check-up height well-baby checkup cancel appointments weighed gaining exam respects ups checks pressured lecture skipped measured trail

3) practitioners respects chiro prescribe gp pocket fp well-baby checkup exam weighed cancel appointments skipped ups checks

significant outliers.

In this network —restricted to contain only nodes in a specific PageRank range— any community detection method can be performed to find dense clusters of contextually co-occurring words. An immediate benefit of running community finding on this network, as opposed to running it on



(a) 1st Context (b) 2nd Context

(c) 3rd Context

Figure 3.13: Bin 9 of three context PageRank profiles in the Mothering.com mentioned in figure 3.12. All three are about doctor visits, but mostly about finding doctors that are not judgemental of non-vaccinating parents. The first one also talks about scheduling office visits significantly.

a subgraph of the original co-occurrence network that only includes nodes in the same PageRank range, is that now we are dealing with a much clearer vision of how words connect in contexts, rather than any random co-occurrence. In other words, words are connected in the new network only if there is a context that includes both of them as significant words, while the global co-occurrence network may have recorded many noisy instances of word pairs coming together in

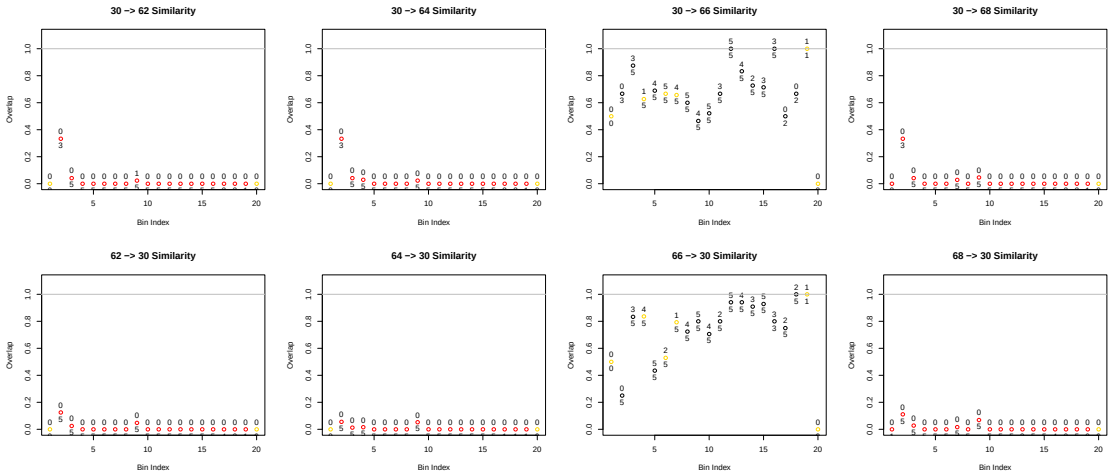


Figure 3.14: Directional bin-wise overlap values for the first two Mothering.com contexts in figure 3.12. The respective indices of the two in dataset contexts are 66 and 30. Other sample contexts with no similarity to 30 are shown for comparison.

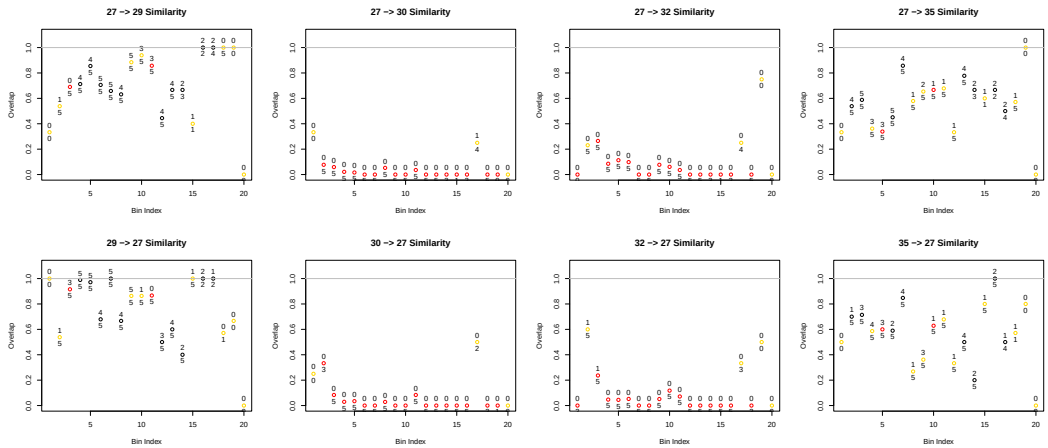


Figure 3.15: Directional bin-wise overlap values between the food-related context 27 in Mothering.com and two other contexts 29 and 35, about food and homeopathy. Other sample contexts with no similarity to 27 are shown for comparison. The core clusters for these contexts are:

`27`: supplements sugar oil vitamins drink organic store liver

`29`: lemon teaspoon capsules grams tastes tsp ascorbic olive tablets capsule gram hylands leaf flax emergenc baking tabs colloidal acidophilus coconut iodine omega spinach herb endotoxin na silica

`35`: herbs remedy herbal homeopath medicines allopathic traditional remedies homeopathy homeopathic

documents without any contextual significance in the co-occurrence. This becomes more of an issue as we go up the PageRank ladder and words become more general and frequent and naturally co-occur more than lower words.

The drawback, however, is that in creating the contextual co-occurrence network we lose information about grouped association of words and only keep pairwise association information. The implication of this information loss is that if word pairs (i, j), (j, k), (i, k) co-occur in three disjoint sets of contexts, all three will have high association and there is no way of making the distinction between them. One strategy to tackle this issue is to only keep connections that imply that one word almost always co-occurs with the other one in context profiles. This results in a directional contextual co-occurrence network in which i is connected to j with weight w'_{ij} if $w'_{ij}/n'_i \geq \alpha$. Where n'_i is the number of contexts word i appears in, and α is a preset threshold around $0.8 \sim 0.9$. Some clusters extracted with this approach, that have arised from contexts mentioned about "doctor visits", are listed below:

1. `parenting wbv measure respected ups gp respects chiropractor respectful exam switch docs doc supportive offices chiro switched friendly`

2. `establish yearly checks asks emergency lecture supports waste pressure drs charts visit appointment scheduled cancel appointments checkup`

3. `holistic pedi tribe finding`

The sharp filtering of smaller association values makes it possible to extract strong conceptual cores, at the cost of losing word combinations that might come together strongly as a group, but do not have strong pairwise association meaning that one word would almost always come with the other. Instead, this method only extracts the very clear groups of words that only co-occur together in contexts and their pairwise contextual association is not vague, which means that it cannot be related to any other possible word combination. The strong filtering of connections thus results in missing some concepts at the middle PageRank layers, as many words tend to participate in

different combinations that make up different concepts as a group. This is in contrast to words of the bottom layer that mostly participate in one group and one context.

Another issue is the hard clustering approach taken in most popular community finding algorithms that does not assign words to more than one cluster. The combination of the filtering problem that loses some connections and the hard clustering issue can be seen in, for example, the cluster above that contains the words "holistic, pedi, tribe, finding". This word group relates to finding "holisitic doctors" and tribal medicine practices, but it misses many other words of the concept such as "doctors".

To extract such patterns we need methods that go beyond pairwise association and take grouped behavior of words into account. Matrix decomposition algorithms, and in particular Non-negative Matrix Factorization (NMF), constitute one such class of methods. NMF algorithms have particularly gained traction recently, as they can be applied in various problems and the non-negativity constraint makes their components easily interpretable. These algorithms decompose a matrix $X \in \mathbb{R}^{m \times n}$ into lower rank representations $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ such that $X \approx WH$, where $k \ll m, n$. When applied directly on a text collection's document-term matrix, NMF is shown to be equivalent to pLSI described in section 2.2.4 [20]. More discussion on the NMF framework and methods can be found in chapter 6.

In our case, the problem can be defined by creating a matrix $R \in \mathbb{R}^{n_c \times n_v}$ that in each row contains the PageRank profile of one context. n_c, n_v are therefore the number of contexts and number of words in our vocabulary respectively, and R_{ij} is the ratio of contextual PageRank to global PageRank for word j calculated as: $\log \frac{PR^{C_i}(j)}{PR(j)}$. We can then factorize this matrix, after some modifications described below, using the NMF framework to get groups of words that mostly rise together and define a concept.

In order to put more emphasis on words that are more representative of a context, we use the idea of Term Frequency-Inverse Document Frequency (TF-IDF) to assign higher values to significant words in a context that do not occur in many other contexts. TF-IDF measures seek to find keywords of a document by evaluating each word's frequency in the document and its frequency in all documents of the corpus. For document set D, a word t, and a document $d \in D$,

if $f_{t,d}$ is the frequency of t in d and $n_{t,D}$ is the number of documents in D that contain t, a popular pair of "term frequency" and "inverse document frequency" functions used to calculate TF-IDF are defined as [42]:

$$TF_{t,d} = \frac{f_{t,d}}{\max_{d \in D} t, d} \tag{3.8}$$

$$IDF_t = \log \frac{|D|}{n_{t,D}} \tag{3.9}$$

Then $TF\text{-}IDF_{t,d}$ is simply equal to the product $TF_{t,d}.IDF_{t,d}$. The "term frequency" measure assigns higher value to words that are more frequent in the current document. The "inverse document frequency" seeks to diminish the importance of common words that occur frequently in every document. The result is that TF-IDF boosts significant words of a document by finding relatively rare words that are frequent in the current document. These words carry the most information about the document's meaning. One may use any other pair of functions that imply the same idea. For instance, another definition of TF and IDF can be written as:

$$TF_{t,d} = 1 + \log f_{t,d} \tag{3.10}$$

$$IDF_t = \log \left(1 + \frac{|D|}{\log n_{t,D}} \right) \tag{3.11}$$

Following this idea to find significant words of each context, we similarly implement a TF-ICF (Term Frequency-Inverse Context Frequency) measure. First, we need to make the matrix R described before non-negative, in order to make its values interpretable as context term intensities and be able to fit it into the non-negative factorization framework. We create a matrix $\hat{R}$, that is a normalized non-negative version of R, by taking the difference between values of R and the median line (drawn in blue in 3.6) and then keeping only non-negative values. In other words, we fit a curve $y = \tilde{m_c}(x)$ to points $(c_{b_i}, m^C_{b_i})$ where c_{b_i} is the center point of bin i, and $m^C_{b_i}$ is the median of context-to-global PageRank ratio values of words in bin i for context C. For the datasets studied a simple second-order polynomial $y = a_2x^2 + a_1x + a_0$ offers a good enough fit, but one could use splines for a more generalized solution [21]. Then for each entry R_{ij} we find $\hat{R}_{ij}$ by:

$$\hat{R}_{ij} = f_r\Big(R_{ij} - \tilde{m_i}\Big(\log PR(j) \Big) \Big) \tag{3.12}$$

The rectifier function $f_r(x) = max(0, x)$ is used to filter out negative values after calculating deviation from the median line.

Given our set of contexts $\mathcal{C}$, and a context $C \in \mathcal{C}$, the term frequency and inverse context frequency measures we calculate using matrix $\hat{R}$ described before are:

$$TF_{t,C} = \hat{R}_{t,C} \tag{3.13}$$

$$ICF_t = \log\left(\frac{|\mathcal{C}|}{\sum_{C\in\mathcal{C}} \hat{R}_{t,C}}\right) \tag{3.14}$$

Therefore, a TF-ICF matrix $T \in \mathbb{R}^{n_c \times n_v}$, that in each row contains TF-ICF value of words for a context, is formed by calculating:

$$T_{ij} = \hat{R}_{ij}.\log\frac{n_c}{\sum_{i=1:n_c} \hat{R}_{ij}} \tag{3.15}$$

Where $n_c = |\mathcal{C}|$ is the number of contexts. T is our context-term matrix containing the memory trace of contexts in its rows as described in section 2.5.1.

In order to find concepts at different PageRank layers, we slide a window of 3 consecutive PageRank bins on words and then focus on the PageRank profile of contexts in this 3-bin layer to find groups of correlated words that rise together. This is done by forming a matrix $T^L \in \mathbb{R}^{n_c \times n_L}$ that contains columns of T corresponding to subset $L \in V$ of words in vocabulary whose PageRank values lie in the 3-bin layer under examination, with $n_L = |L|$. This matrix can then be decomposed into two non-negative matrices $W^L \in \mathbb{R}^{n_c \times k_L}$ and $H^L \in \mathbb{R}^{k_L \times n_L}$ such that $T^L \approx W^L H^L$. In this decomposition, rows of H^L become a basis that can reconstruct a context's layer TF-ICF vector (a row i in T^L) by a linear combination using coefficients stored in the respective row i of W^L. The low-rank compression forces groups of words that come up together to have simultaneously high values in some row of H^L. Therefore, finding high values in rows of H^L can give us word groups that represent available concepts in this layer. After extracting concepts at each layer, the layer window is shifted one bin to the right and factorization is repeated. If the PageRank spectrum is divided into 20 bins, we will then have 18 different decompositions in bin windows (1,2,3), (2,3,4), ... , (18,19,20).

The top words of each row in H^L, sorted by the words' value in the row, are given in tables 3.3, 3.4, 3.5 for one of the middle layers in the Mothering.com dataset. The choice of number of clusters k_L in each layer can be a topic of further research. For this dataset, we have set $k_L = 50$ in all layers, which provides a diverse set of clusters in most middle layers. Ideally, as we go up the PageRank spectrum, the number k_L could be decreased to account for smaller number of concepts in higher layers of the hierarchy which contain more general and inclusive concepts that each may relate to many lower layer clusters. The study of the effects of changing k_L as we go higher in the PageRank spectrum is left for future work.

Some examples of the clusters related to previously discussed contexts are clusters 5 and 37 that relate to "doctor visits" and "finding alternative medicine doctors" respectively, bringing up each concept's words in the focus layer. Clusters 28, 34, and 43 are also examples of how NMF's analysis of group co-occurrence, as opposed to pairwise co-occurrence, can extract different combinations of words regarding vaccine ingredients thimerosal and aluminium that refer to different concepts. Cluster 28 is about manufacturer vaccine products that contain thimerosal, 34 refers to discussions on toxicity of thimerosal, and 43 is about the use of fetal cells in vaccines which can become a religious issue.

3.5.2 Hierarchical Structuring

Once we have concepts of every layer window extracted, we would like to find how these concepts are structured across the PageRank spectrum. To devise a strategy to build a hierarchical organization of these layer clusters, it is useful to point out how clusters across different layers relate to each other. For any cluster extracted at a layer L_i on bins $(i, i+1, i+2)$, there are four possible cases on how it relates to clusters of the previous layer L_{i-1}:

1. The cluster is a continuation of a cluster previously extracted at the lower layer L_{i-1} on bins $(i-1, i, i+1)$. In this case most of the high-valued words of the given cluster and its lower-layer counterpart are similar over the shared bins $(i, i+1)$.

2. The cluster is formed by a merge happening on two or more clusters at the lower layer L_{i-1}.

	Top 10 Words of Layer Cluster
Cluster 1	meningitis strains bacterial prevnar strain pneumonia viral ear types carry
Cluster 2	crying sleep bed hour minutes er morning tired seizures pain
Cluster 3	influenza strain media deadly strains spreading killed dead mass prevention
Cluster 4	military base force approved members member policy air opportunity writing
Cluster 5	visits appointment peds appt insurance supportive clinic pediatricians drs phone
Cluster 6	series ipv dtp booster diphtheria diptheria prevnar boosters skip efficacy
Cluster 7	belief court held practices religion explanation offered act beliefs member
Cluster 8	hot slightly throat bed straight circumstances stuck useless rid drink
Cluster 9	introduced smallpox mortality incidence outbreaks epidemic united statistics million decades
Cluster 10	girl watch whooping syndrome watching babe arm hurt newborn injection
Cluster 11	canada mandatory rights force grade forced move outbreak act entry
Cluster 12	scientific sides claims debate sources industry conclusion published pharmaceutical compare
Cluster 13	company pay cost paid pharma pharmaceutical industry market business fda
Cluster 14	nutrition environment bodies toxic humans nature modern genetic conditions factors
Cluster 15	guide spite raise pro inserts confident sides pages resources stats
Cluster 16	power reality man front society buy saved hit sit walk
Cluster 17	referring accurate surely reasonable percentage logical relevant statements individuals disagree
Cluster 18	rash red nose raised throat arm hot viral morning noticed
Cluster 19	vitamin liver organic drink vit foods milk store eating daily
Cluster 20	rotavirus market oral approved fda trials recommendation vaers events meeting

Table 3.3: Layer clusters 1 to 20 out of 50 clusters extracted using the NMF method on a window over PageRank bins (11,12,13) of Mothering.com dataset.

	Top 10 Words of Layer Cluster
Cluster 21	germs touch clean colds playing stronger store air careful healthier
Cluster 22	tv watching watch media piece pharma um ridiculous stories hmmm
Cluster 23	breastmilk breast milk bf breastfed formula fed nursing feed gut
Cluster 24	journal clinical researchers incidence percent published primary trials development increased
Cluster 25	www sites dangers pm reference anti-vax print google suffered resources
Cluster 26	national meeting thimerosal review american published media hearing fda researchers
Cluster 27	edited vaxxing answered intended choosing anti-vax originally vpd pro-vax boards
Cluster 28	insert package thimerosal inserts influenza product amounts pediatric hepatitis listed
Cluster 29	department immunizations signed section services copy request stating dept apply
Cluster 30	seizures vaers syndrome reports events permanent loss pain crying reporting
Cluster 31	route initial persons inside efficacy preventing transmission treatments consequences infectious
Cluster 32	threads posting boards mothering mt mdc member message members reply
Cluster 33	court legal consent rights force action forced rules injury door
Cluster 34	aluminum amounts thimerosal injecting toxic tissue inject animal cells animals
Cluster 35	autistic disorder connection genetic neurological disorders diagnosis special damaged development
Cluster 36	cp chickenpox varicella wild younger naturally booster rash contagious sister
Cluster 37	homeopathic md alternative healing pediatricians meds treating wonderful treatments surgery
Cluster 38	transmission contracting infected wild protected contract contracted antibodies herd outbreaks
Cluster 39	pregnancy negative tests mothers girls positive antibodies affect hospitals routine
Cluster 40	mo mos youngest till boys oldest yr yrs younger partially

Table 3.4: Layer clusters 21 to 40 out of 50 clusters extracted using the NMF method on a window over PageRank bins (11,12,13) of Mothering.com dataset.

	Top 10 Words of Layer Cluster
Cluster 41	throat nose pneumonia respiratory colds asthma whooping ear meds stomach
Cluster 42	allergy allergic skin milk asthma foods products meds eating autoimmune
Cluster 43	fetal tissue cell cells lines animal hepatitis inserts aluminum package
Cluster 44	travel places areas city clean america lived parts access oral
Cluster 45	parenting conversation respect listen shut ignorant educate educated convince anti-vax
Cluster 46	deep clean cut injury inside environment animal animals typically skin
Cluster 47	hospitals newborn eye vit nurses consent staff routine pregnancy refused
Cluster 48	copy letter fill forms signed stating grade preschool pm request
Cluster 49	schools grade preschool attend private class education requirements entry center
Cluster 50	philosophical religion laws private beliefs schools file forms attend fill

Table 3.5: Layer clusters 41 to 50 out of 50 clusters extracted using the NMF method on a window over PageRank bins (11,12,13) of Mothering.com dataset.

When a merge occurs, the current cluster will include words from its multiple lower-layer corresponding clusters over the shared bins $(i, i+1)$.

3. The cluster is formed by a division occurring on a cluster in lower layer L_{i-1} or a re-arrangement of grouping on some of the lower layer clusters. In such cases, the mentioned cluster only partly overlaps with some clusters on layer L_{i-1}.

4. This is a new cluster appearing at this layer which has not shown up in lower layers.

This means that concepts emerge at some range of the PageRank spectrum, evolve when going up the PageRank ladder by either continuing, merging with other clusters, and dividing into different clusters, and then possibly vanish at some point. We can build up a hierarchy of concepts by studying the evolution of clusters throughout the PageRank spectrum. One approach to building up such a hierarchical structure is examining the overlap of clusters from one layer to the next.

In order to find how a cluster u at layer L_i evolves in the next layer L_{i+1}, we first retrieve its respective row vector $H^{L_i}_{u,\star}$. Assuming $B_{i+1,i+2}$ is the set of words in PageRank bins b_{i+1} and b_{i+2}, let $h_u^{L_i,overlap}$ be a vector composed of elements of $H^{L_i}_{u,\star}$ that correspond to words in the overlap set $B_{i+1,i+2}$. Similarly, for any cluster v at layer L_{i+1} on bins $(i+1, i+2, i+3)$, we form a vector $h_v^{L_{i+1},overlap}$ that is created of elements of $H^{L_{i+1}}_{v,\star}$ that correspond to words in the overlap set $B_{i+1,i+2}$ with the same order as in $h_u^{L_i,overlap}$,. Therefore, $h_u^{L_i,overlap}$ and $h_v^{L_{i+1},overlap}$ are vectors of the same length that contain, with similar ordering, the values assigned to word set $B_{i_1,i+2}$ by u and v respectively. We then try to create a linear reconstruction of the normalized version of $h_u^{L_i,overlap}$ by normalized versions of vectors $h_v^{L_{i+1},overlap}$ for $v = 1...k_L$ such that:

$$\hat{h}_u^{L_i,overlap} = \frac{h_u^{L_i,overlap}}{\left\| h_u^{L_i,overlap} \right\|_1} \tag{3.16}$$

$$\hat{h}_v^{L_{i+1},overlap} = \frac{h_v^{L_{i+1},overlap}}{\left\| h_v^{L_{i+1},overlap} \right\|_1} \tag{3.17}$$

$$\hat{h}_u^{L_i,overlap} \approx \sum_{v=1}^{k_L} \alpha_v \hat{h}_v^{L_{i+1},overlap} \tag{3.18}$$

The mapping from $h_u^{L_i,overlap}$ to vectors $h_v^{L_i,overlap}$ is thus stored in coefficients α_v. The linear reconstruction problem can be solved by the regression problem of minimizing $L2$-norm between $\hat{h}_u^{L_i,overlap}$ and $\tilde{h}_u^{L_i,overlap} = \sum_{v=1}^{k} \alpha_v h_v^{L_{i+1},overlap}$:

$$\min_{\alpha \in \mathbb{R}^{k_L}} \left\| \hat{h}_u^{L_i,overlap} - \sum_{v=1}^{k_L} \alpha_v \hat{h}_v^{L_{i+1},overlap} \right\|^2 \tag{3.19}$$

We are more interested in sparse versions of $\alpha \in \mathbb{R}^k$ that can map similar topics with high coefficients and are robust to slight differences in distribution of values between $\hat{h}_u^{L_i,overlap}$ and $\hat{h}_v^{L_{i+1},overlap}$. This can be achieved by solving an $L1$-regularized regression problem that minimizes [24]:

$$\min_{\alpha \in \mathbb{R}^{k_L}} \left(\left\| \hat{h}_u^{L_i,overlap} - \sum_{v=1}^{k_L} \alpha_v \hat{h}_v^{L_{i+1},overlap} \right\|^2 + \|\alpha\|_1 \right) \tag{3.20}$$

Adding the $L1$-norm of coefficient vector α to the cost function is a well-known technique to help find sparse representations of the linear reconstruction model [23].

Figure 3.16 shows a distant plot of the hierarchical structure built using regression mapping on overlapping bins of layered NMF clusters. Each node represents an NMF cluster and an edge is placed between cluster u of layer L_i and cluster v of layer L_{i+1} with weight equal to α_v calculated above. The horizontal level of a node shows its layer in the hierarchy, so nodes at the same level correspond to various NMF clusters extracted in one layer. It is noticeable how concepts form connected chains of layer clusters across PageRank layers. These chains mostly have a tendency to merge as we go up the PageRank scale, nevertheless they may also divide and connect to various clusters in the higher layers, which means the connected components of our hierarchy do not necessarily have a tree structure. This is in contrast with models like the hierarchical variation of LDA [22] that assume a tree of topics as the backbone of generative model of documents.

An important observation can be made, for instance, from the close-up of a piece of the hierarchy in figure 3.17, which exhibits the role of vaccine ingredients and their presumed link to autism. The lowest layer cluster shown here on "thimerosal", extracted from layer L_{12} on bins $(12, 13, 14)$, can be seen connecting to two different chains in the higher layer, one about "doses" of these ingredients, and the other one about "autism and brain damage".

Our other example concepts mentioned before can be viewed in the hierarchy cut in figure 3.18 that again shows a piece of the hierarchy in layers L_{12} to L_{15}. One can see how "scheduling doctor appointments" has its own chain in the hierarchy, along with another chain that relates to finding "alternative medicine doctors". These two chains merge at layer L_{14} and beyond, where they share most words on "doctor visits".

Figure 3.16: Hierarchical structure of the layered NMF clusters extracted from forums of Mothering.com. Very small values of edge weights below 0.3 are filtered out for better visualization.

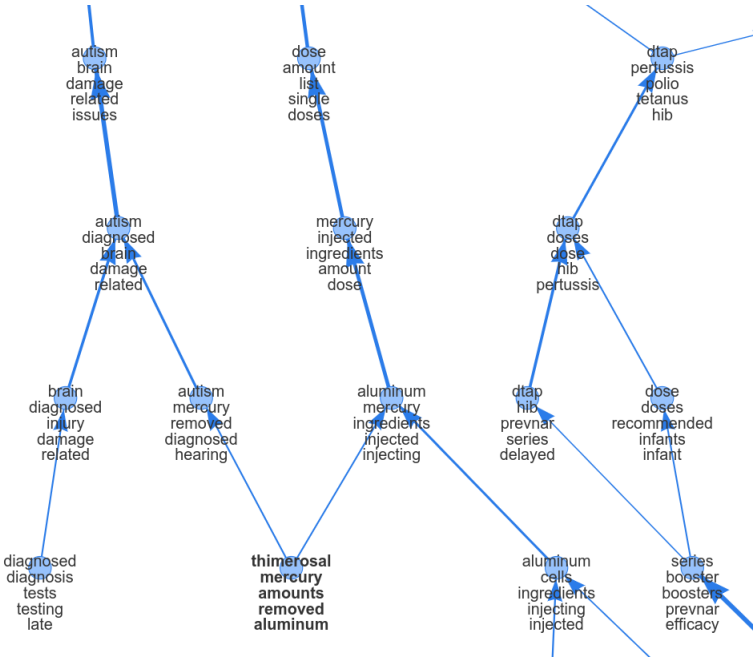


Figure 3.17: A crop of the hierarchical structure of NMF clusters showing some clusters on "vaccine ingredients" and "autism" in the four layers ranging from L_{12} (bottom nodes) on bins $(12, 13, 14)$ up to L_{15} (top nodes) on bins $(15, 16, 17)$. Top 5 words of each cluster are shown on its respective node.

Figure 3.18: A crop of the hierarchical structure of NMF clusters showing some clusters on "scheduling doctor appointments" and "homeopathy and alternative medicine" in the four layers ranging from L_{12} (bottom node) on bins $(12, 13, 14)$ up to L_{15} (top node) on bins $(15, 16, 17)$.

3.6 Summarization

3.6.1 Summary Topics

An attractive property of LDA is the fact that one can use it on a large corpus of text to get a quick summary of topics available in the text collection. By doing so, a dataset of tens of thousands of documents are reduced to perhaps 20 topics, and the top 10-20 words of each topic can be glanced over for a sense of what are the major ideas being discussed in the documents.

As indicated earlier, our set of bottom clusters are representative of document groups that are very similar in context and thus very similar in their overall composition of words. These bottom clusters contain all topical information of the corpus compressed in their PageRank profiles. Therefore, we can use their PageRank profiles to create a summary of the corpus comparable to LDA topics. This is done by factorizing the context-term matrix $T \in \mathbb{R}^{n_c \times n_v}$ described in

previous sections, which in each row contains significance of terms in one context. Using the NMF framework, we find matrices W and H such that:

$$T^{n_c \times n_v} \approx W^{n_c \times k} H^{k \times n_v} \tag{3.21}$$

A row of H can now be regarded as a topic, with high-value words of the row being more significant words of the the topic it contains. The set of rows in H thus provide k topics that summarize the corpus.

We use a weighted version of the NMF problem that, given a weight matrix $M \in \mathbb{R}^{n_c \times n_v}$, minimizes weighted Frobenius norm of the difference between T and its reconstruction WH with the following cost function:

$$\min_{W \in \mathbb{R}^{n_c \times k}, H \in \mathbb{R}^{k \times n_v}} \sum_{i=1}^{n_c} \sum_{j=1}^{n_v} \left\{ M_{ij} (T_{ij} - (WH)_{ij})^2 \right\} \tag{3.22}$$

One can use different weighting strategies to bias the reconstruction problem towards certain contexts or certain words. For instance, it is possible to estimate the number of documents N^C related to each context and use this value on the respective row of T. One estimate for N^C can be the median of all co-occurrence values between pairs of words inside the core cluster of context C. A weight matrix M_D that incorporates estimated document counts of each context can be formed as:

$$M_D^{n_c \times n_v} = \begin{bmatrix} N^{C_1} & \cdots & N^{C_1} \\ N^{C_2} & \cdots & N^{C_2} \\ \vdots & \ddots & \vdots \\ N^{C_{n_c}} & \cdots & N^{C_{n_c}} \end{bmatrix} \tag{3.23}$$

Such weighting scheme implies that contexts with higher document count will have more influence on output topics in rows of H. As a consequence, topics associated with these contexts are more likely to be "picked up" in the summary.

Another weight matrix M_P that assigns higher weight to more frequent words can be created

using PageRank value of words as a measure of their importance in the text corpus:

$$M_D^{n_c \times n_v} = \begin{bmatrix} PR(1) & \cdots & PR(n_v) \\ PR(1) & \cdots & PR(n_v) \\ \vdots & \ddots & \vdots \\ PR(1) & \cdots & PR(n_v) \end{bmatrix} \tag{3.24}$$

This weight matrix assures that higher-PageRank words, which are of central value in a topic, have higher influence on the choice of topics. In other words, two medium-Pagerank clusters of words will likely not be combined in a row of H if, in the context PageRank profiles, they appear with non-similar high-PageRank clusters.

In addition, the more pronounced presence of high-PageRank words make output topics more coherent by providing the more general context that other more detailed topic words have appeared in. In fact, to accentuate the presence of higher-PageRank words and account for smaller PageRank boost in higher layers of the PageRank profile, we also apply a transformation on the resulting matrix H. That is, we re-order words by applying a multiplicative factor, equal to PageRank of words, to their respective entries in the H matrix, before sorting each row to get top words of the topic. Therefore, the matrix whose sorted rows give us top topic words is:

$$H_t^{k \times n_v} = \begin{bmatrix} PR(1)H_{11} & \cdots & PR(n_v)H_{1n_v} \\ PR(1)H_{21} & \cdots & PR(n_v)H_{2n_v} \\ \vdots & \ddots & \vdots \\ PR(1)H_{k1} & \cdots & PR(n_v)H_{kn_v} \end{bmatrix} \tag{3.25}$$

The results of this summarization method on Mothering.com forums for $k = 20$ topics can be viewed in table 3.6. The topics that have formed around our sample contexts previously discussed on "vaccine ingredents" are 2 and 16. Topic 2 is on the use of fetal cells in vaccines and 16 is about the dialogue on the mercury-autism link. The discussions around "doctor appointments" and "finding alternative doctors" have merged into one topic, numbered 11 in the table.

As one increases the number of topics, more detailed topics of discussion start to show up. This is evident in table 3.7 that shows two different topics extracted on "doctor appointments" and "alternative medicine" when we extract $k = 60$ topics.

For comparison, LDA results for the same number of topics $k = 20$ are shown in table 3.8.

	Top 10 Words of Topic
Topic 1	books book guide thoughtful mendelsohn websites spite links cave reading
Topic 2	ingredients aluminum cells tissue formaldehyde fetal cell aborted mercury human
Topic 3	birth hospital midwife born newborn baby eye pregnancy ob delivery
Topic 4	force care pay medical court rights order legal forced call
Topic 5	school exemption religious form state exemptions required schools law letter
Topic 6	study studies results clinical response incidence journal data rate increase
Topic 7	travel countries country polio africa india traveling places areas water
Topic 8	immunity adults exposed adult measles childhood cases mumps virus disease
Topic 9	science argument scientific true evidence fact agree world point opinion
Topic 10	fever symptoms days pain hours vomiting cold sleep night infection
Topic 11	doc ped visits visit office pediatrician doctor docs appointment peds
Topic 12	decisions choices talk conversation decision parenting telling discuss respect choice
Topic 13	government national companies fda drug public pharma safety committee money
Topic 14	sick colds germs cold kid healthy home hands breastfed house
Topic 15	tetanus wound deep clean wounds nail cut puncture horse blood
Topic 16	autism autistic disorder problems spectrum brain damage diagnosed speech mercury
Topic 17	meningitis hib bacteria strains prevnar invasive pneumococcal replacement bacterial serotype
Topic 18	rash red allergy allergies reaction allergic worse fever skin started
Topic 19	foods diet food milk eat vitamin oil vitamins organic eating
Topic 20	flu news people pandemic bird swine hype media influenza strain

Table 3.6: Summary topics extracted for forums of Mothering.com using the weighted NMF method on context PageRank profiles. The number of topics is set to $k = 20$.

	Top 30 Words of Topic
NMF60 Topic 33	ped visit visits doc office pediatrician appointment peds doctor docs insurance appt practice wbv check asked vaxing checks cps call friendly pedi practitioner supportive nurse pediatricians pressure fp told clinic
NMF60 Topic 45	homeopathic md alternative homeopathy remedies medicine chiropractor homeopath chiro treat healing holistic treatment allopathic family naturopath herbs homeopaths doctors dr area finding chiropractic chiropractors doc natural care practice naturopaths heal

Table 3.7: Two topics related to interactions with doctors in $k = 60$ NMF topics. These topics are combined in topic 11 of table 3.6 when we extract $k = 20$ topics.

	Top 10 Words of Topic
Topic 1	hep baby hospital test birth blood tb shot pregnancy pregnant
Topic 2	measles pox chicken cp immunity rubella mumps shingles mmr immune
Topic 3	doctor ped doc dr doctors office insurance practice visit visits
Topic 4	tetanus wound rabies dog shot cat dogs cats wounds vet
Topic 5	pertussis cough sick kids whooping vaxed baby immune months unvaxed
Topic 6	people article thread read news lol love story post interesting
Topic 7	mercury aluminum vaccines thimerosal cells vaccine ingredients formaldehyde human aborted
Topic 8	vitamin milk diet food eat water immune foods system body
Topic 9	vax feel research decision dh family kids vaccinate vaxing vaxed
Topic 10	religious exemption religion beliefs state letter law philosophical school exemptions
Topic 11	hpv vaccine cancer cervical fda money drug bill gardasil companies
Topic 12	polio smallpox countries travel country water disease opv virus world
Topic 13	flu shot vaccine influenza swine year virus pregnant pandemic season
Topic 14	school form state exemption religious immunization law exemptions schools required
Topic 15	vaccine cases disease meningitis hib deaths reported infection risk pertussis
Topic 16	autism study children autistic studies vaccines mercury mmr disorders sids
Topic 17	fever reaction reactions rash days tylenol normal ear seizures pain
Topic 18	dtap hib months shots schedule vax mmr hep prevnar vaccines
Topic 19	book research read info information books vaccines vaccine links good
Topic 20	people vaccines disease doctors research vaccinate immune science evidence medical

Table 3.8: LDA topics extracted for Mothering.com forums with $k = 20$ topics. The words of each topic are ordered according to their z-score, instead of the actual probability values in multinomial distributions stored in rows of β. This ordering helps bring up more contextually important words of each topic, and suppresses more general words that naturally have high values in topic distributions that are trying to reconstruct distribution of words in documents.

3.6.2 Summary Structures

The hierarchical network of concepts extracted in section 3.5 provides a structural view of the knowledge base one can learn from a text corpus. Any document touches on parts of this hierarchy and follows some paths of mostly connected concepts in order to provide coherent information. For any of our context cores, which represent a group of similar documents, we can find layer clusters they incorporate at a layer L by examining their corresponding row in matrix W^L. Let W^L_{c*} be the row of W corresponding to bottom context cluster C. This row contains the coefficients used for a linear reconstruction of context C's layer TF-ICF vector stored in $T^L_{c,\star}$, using layer clusters that are rows of H^L. In other words:

$$T^L_{c,\star} \approx \sum_{v=1}^{k_L} W^L_{cv} H^L_{v,\star} \tag{3.26}$$

We can find layer clusters activated in the PageRank profile of C by finding clusters v that have a high coefficient W^L_{cv}. Given a high enough number of clusters k_L that would provide a clear separation of concepts at layer L (so that layer clusters are not formed by a combination of the underlying dense cores of this layer and each cluster is formed around only one core), most document groups will be created by a very sparse combination of these layer concepts. Figure 3.19 contains a few examples of values in row vectors of W^L that show the distribution —or the mapping— of a context over layer clusters. As seen in these examples, most contexts make use of only very few of the layer clusters for creating their PageRank profile. Each row $W^L_{c\star}$ can be normalized by its sum of values, to find relative importance of each layer cluster in context C's composition:

$$\hat{W}^L_{cv} = \frac{W^L_{cv}}{\sum_{v=1}^{k_L} W^L_{cv}} \tag{3.27}$$

When these mappings at each layer for a given context are superimposed on the hierarchical structure, one can see the concepts at different layers that are invoked in a certain context. Figure 3.20 is an instance of such a mapping onto the hierarchy for a context related to nutrition and treatment of colds. Even though the mapping is done independently on each layer, it is evident how the context mostly evokes continuous chains of concepts that would provide coherence.

In addition to maps of contexts onto the concept hierarchy, an interesting question would be:

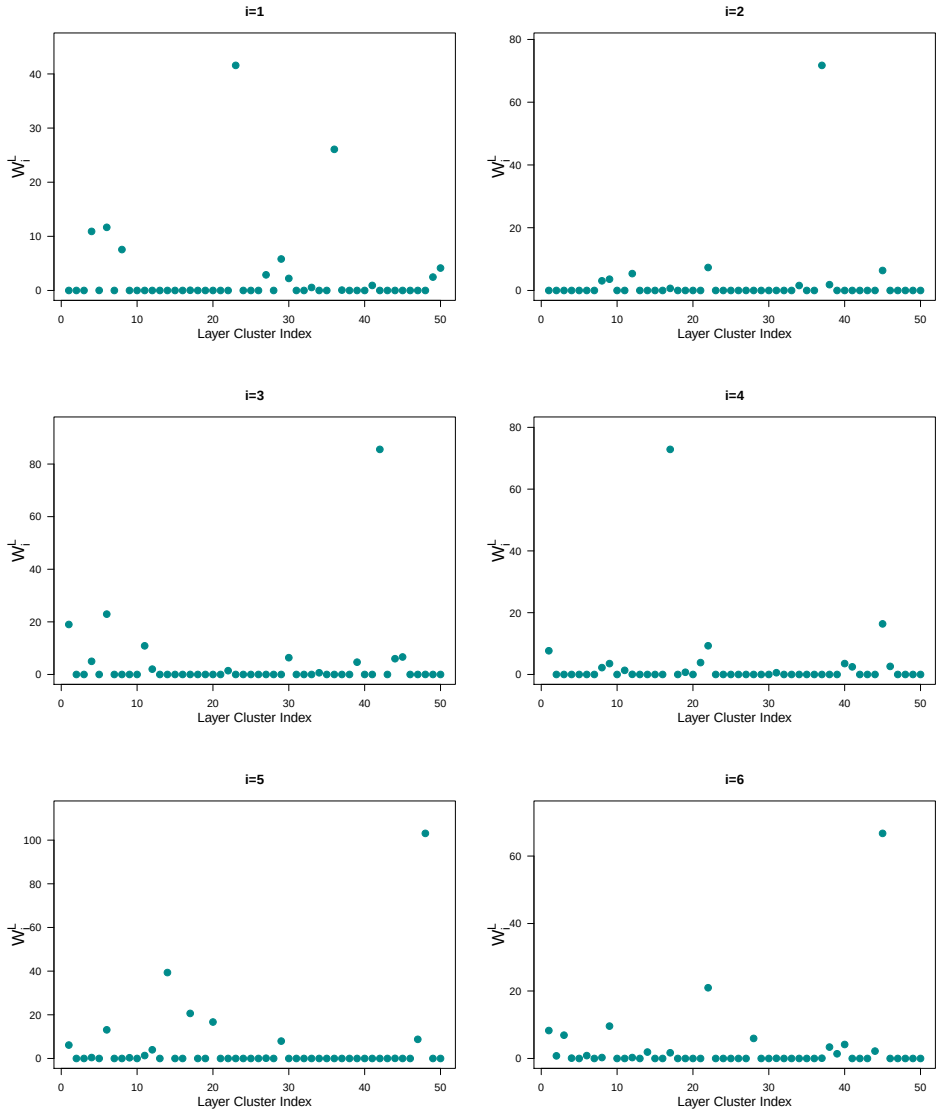


Figure 3.19: Row vectors of matrix W^L that contain a context's mapping onto layer clusters. The vectors for 6 different Mothering.com contexts are shown in middle layer L_{10} which spans over bins $(10, 11, 12)$.

what are the most frequently used paths and combinations of concepts in this structure when writing documents? To rephrase, beyond the word combination summaries provided in section 3.6.1, is it possible to show structural summaries of the text corpus?

This can be achieved with a mapping of topic summaries extracted by factorization of the entire context-term matrix T, which have most significant word combinations of the corpus, onto

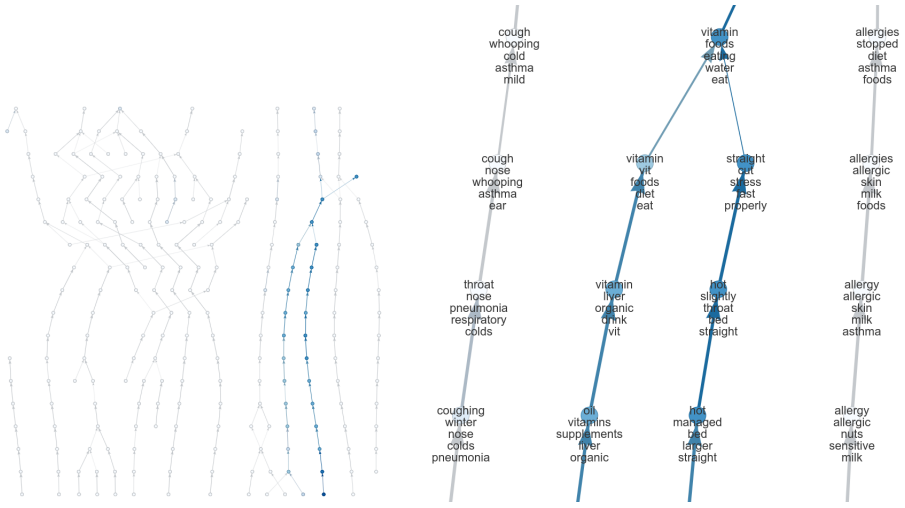


Figure 3.20: Mapped structure of a context onto the concept hierarchy. Normalized mapping values $\hat{W}^L_{cv}$ from context c to layer clusters v are encoded in node colors, with darker colors assigned to higher values. A distant view of the mapped highlighted chain is shown on top, and on the bottom a close-up of some mapped layer clusters. This context is about food and home remedies, built around the following core cluster:

• lemon teaspoon ascorbic capsules grams tastes kefir tsp mgs gram omega emergen-c hylands leaf capsule absorption flax olive coconut kg iu tablets b12 diarrhoea baking dosages acidophilus herb spinach iodine tabs copper migraines cap cabinet folic

the hierarchy of concepts. An approach to finding such mappings is to inspect how words' assigned values in a topic, that is a row of H in the summary factorization problem, match with their corresponding values in layer clustering matrices H^L. To that end, for each layer, we would need to find the subset of matrix H that contains the layer words. Assuming word indices are ordered by their PageRank value, this translates into a block of the matrix:

$$H^{k \times n_v} = \begin{bmatrix} H_{11} & \cdots & \overbrace{H_{1l_1}}^{\text{L}} & \cdots & H_{1l_{|L|}} & \cdots & H_{1n_v} \\ H_{21} & \cdots & H_{2l_1} & \cdots & H_{2l_{|L|}} & \cdots & H_{2n_v} \\ \vdots & & \vdots & & \vdots & & \vdots \\ H_{k1} & \cdots & H_{kl_1} & \cdots & H_{kl_{|L|}} & \cdots & H_{kn_v} \end{bmatrix} \tag{3.28}$$

Where l_1 through $l_{|L|}$ are the indices of the lowest-PageRank and highest-PageRank words of layer L. Let's call this block of the matrix H^{GL}, which contains values assigned to layer L's words

in the summarization NMF output. For a summary topic u, we take the respective row $h_u^{GL} = H_{u,\star}^{GL}$ and use the sparse regression framework, similar to cross-layer mappings of layer clusters, in order to find its closest matching row vectors $h_v^L = H_{v,\star}^L$ in layered NMF clusters. This is done by first normalizing the vectors by their $L1$-norm and then solving the regression problem as follows:

$$\hat{h}_u^{GL} = \frac{h_u^{GL}}{\left\|h_u^{GL}\right\|_1} \tag{3.29}$$

$$\hat{h}_v^{L} = \frac{h_v^{L}}{\left\|h_v^{L}\right\|_1} \tag{3.30}$$

$$\hat{h}_u^{GL} \approx \sum_{v=1}^{k_L} \alpha_v \hat{h}_v^L \tag{3.31}$$

The cost function of the regression problem would then be:

$$\min_{\alpha \in \mathbb{R}^{k_L}} \left(\left\| \hat{h}_u^{GL} - \sum_{v=1}^{k_L} \alpha_v \hat{h}_v^L \right\|^2 + \|\alpha\|_1 \right) \tag{3.32}$$

Which after minimization would give us a mapping from topic u to its related layer clusters v, provided in coefficients α_v. A similar mapping is calculated in every layer to retrieve topic u's trace in the concept hierarchy.

Analyzing each topic u as described above would yield to an overlay that highlights parts of the hierarchy. The set of k summary topics together then give us k such structures that are the most significant hierarchical signatures in our dataset's context PageRank profiles. As demonstrated in figure 3.21 for an example summary topic, these structural summaries also mostly follow consistent and well-connected paths upward the hierarchy.

The mapping approach used for finding each summary topic's hierarchical signature is in fact so generalized that any topic —extracted by any means— presented as a distribution of values over words of the vocabulary can be mapped onto the hierarchy with a similar framwork. This includes LDA topics contained in row vectors of the β matrix as multinomial distributions of words. Similarly, on any layer L, one can find the respective block of the β matrix:

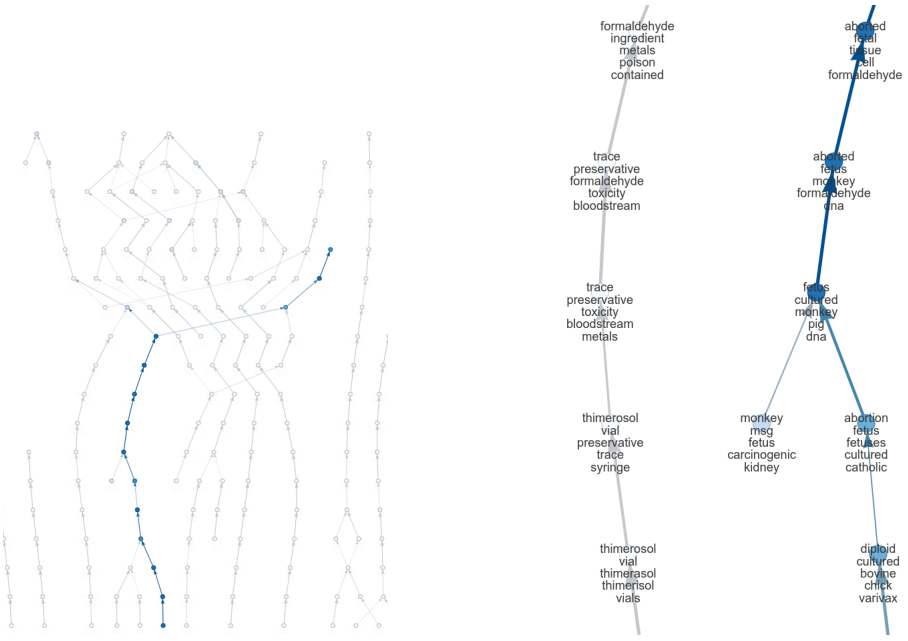


Figure 3.21: A distant view (left) and close-up (right) of the mapped structure of a summary topic onto the concept hierarchy. Darker node colors depict higher mapping coefficients α_v in the solution of the sparse regression problem. The topic, extracted from Mothering.com by setting $k = 60$ summary topics, is about the use of fetal cells in vaccine production and the top 10 words of the topic are:

• aborted fetal cells tissue fetus ingredients lines cell cultured monkey

$$\beta^{k\times n_v} = \begin{bmatrix} \beta_{11} & \cdots & \overbrace{\beta_{1l_1}}^{\text{L}} & \cdots & \beta_{1l_{|L|}} & \cdots & \beta_{1n_v} \\ \beta_{21} & \cdots & \beta_{2l_1} & \cdots & \beta_{2l_{|L|}} & \cdots & \beta_{2n_v} \\ \vdots & & \vdots & & \vdots & & \vdots \\ \beta_{k1} & \cdots & \beta_{kl_1} & \cdots & \beta_{kl_{|L|}} & \cdots & \beta_{kn_v} \end{bmatrix} \tag{3.33}$$

Then in the block β^L formed of β values for layer words, each topic u's distribution values over the layer words $\beta^L_u = \beta^L_{u,\star}$ can be normalized and mapped onto layer clusters:

$$\hat{\beta}^L_u = \frac{\beta^L_u}{\|\beta^L_u\|_1} \tag{3.34}$$

$$\tilde{\alpha} = \underset{\alpha\in\mathbb{R}^{k_L}}{\arg\min} \left(\left\| \hat{\beta}^L_u - \sum_{v=1}^{k_L} \alpha_v \hat{h}^L_v \right\|^2 + \|\alpha\|_1 \right) \tag{3.35}$$

Which again is performed over all layers L_i to find hierarchical representation of an LDA topic u. Figure 3.22 displays the result of such mapping for one LDA topic. The continuity of mapped structures for LDA topics also serves a validation for the extracted hierarchical structure of concepts in the corpus.

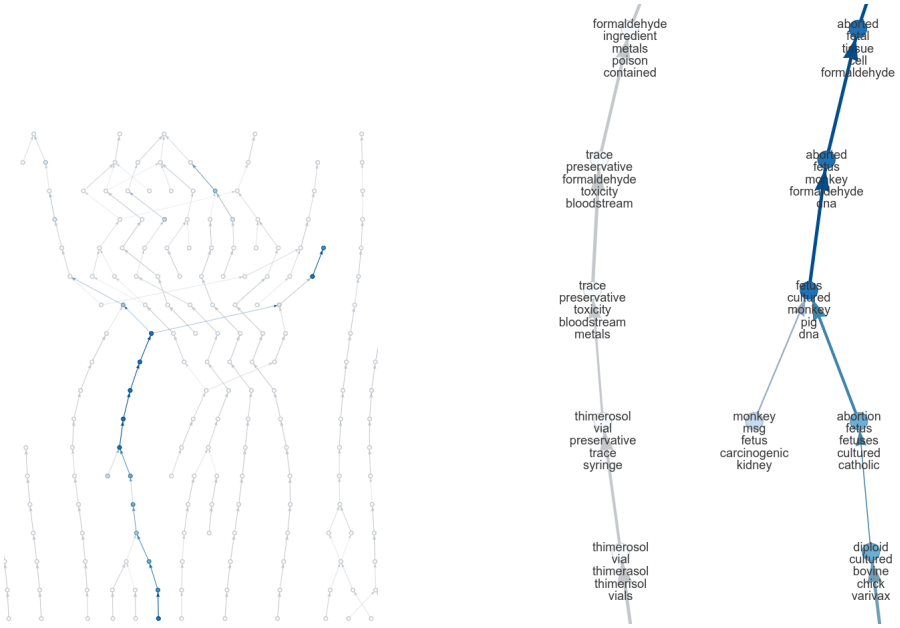


Figure 3.22: A distant view (left) and close-up (right) of the mapped structure of an LDA topic onto the concept hierarchy. Darker colors show higher mapping coefficients. This topic is again on "fetal cells" extracted from Mothering.com forums, setting LDA number of topics $k = 20$. The top 10 words for this topic are:

• aborted fetal cells tissue cell dna human monkey abortion

3.7 Document Mapping

Another useful feature is to map given documents onto the concept hierarchy to find the document's structural composition of concepts. In the bag-of-words model, a document is essentially described as a normalized frequency vector over vocabulary words $(\omega_1, \omega_2, ..., \omega_{n_v}), s.t. \sum_i \omega_i = 1$, along with the number of words in the document N_d. The frequency of each word in the document is then given by $N_d\omega$.

The document's distribution over the vocabulary can be mapped onto the hierarchical structure with a similar strategy as LDA topic distributions. On each layer L the mapping on layer clusters is given by:

$$\tilde{\alpha} = \underset{\alpha \in \mathbb{R}^{k_L}}{\arg\min} \left(\left\| \omega^L - \sum_{v=1}^{k_L} \alpha_v \hat{h}_v^L \right\|^2 + \|\alpha\|_1 \right) \tag{3.36}$$

Where ω^L is formed from elements of ω corresponding to words in layer L. A sample structural mapping for a document is shown can be seen in figures 3.23 and 3.24.

Figure 3.23: A distant view (left) and close-up (right) of the mapped structure of a document onto the concept hierarchy. Darker colors show higher mapping coefficients. This document is a Mothering.com thread on "food and supplements":

... Elimination diets and definately not Rotation Diets like she suggests, by themselves do not heal the gut. And then, you have to supplement to replace all the food you are taking out (and supplements are not as effective as food based nutrients and also can be damaging to a leaky gut b/c they are artificial.) My DS has been on minerals and special diet since Sept. ...

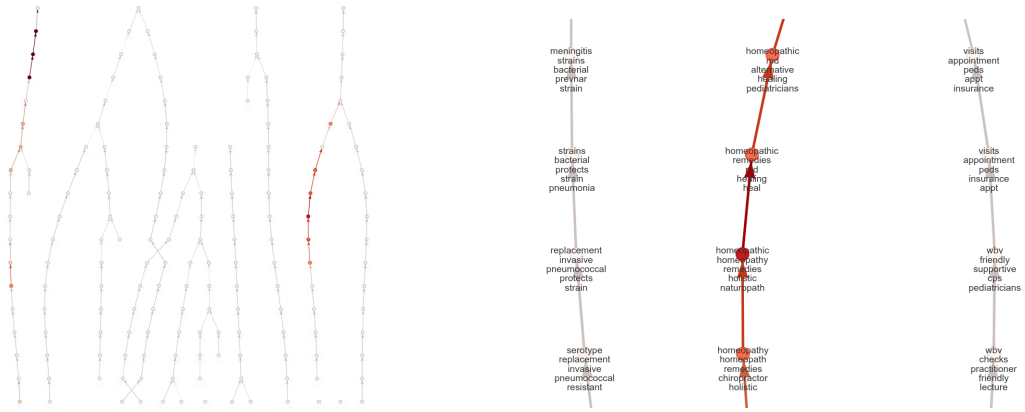


Figure 3.24: A distant view (left) and close-up (right) of the mapped structure of a document onto the concept hierarchy. Darker colors show higher mapping coefficients. This document is a Mothering.com thread on "homeopathy":

... Lydall's book is more about alternative treatments vs lots of specific vax info. I found this book to be so poorly done... While I'm not really into homeopathy, her book is written from the perspective of someone who has actually treated these disease, so she's able to breakdown what to expect in a normal course of illness. ...

CHAPTER 4

Experimental Results and Evaluation

4.1 Datasets

We have run experiments on seven different datasets that represent different sources of text data. The first and second datasets contain threads from online forums in two public discussion websites *Mothering.com* [1] and *Cafemom* [25] related to mothers' issues, their Q&A and conversations about vaccination. More specifically, this dataset follows the conversations on vaccination and concerns of a group of the public over its adverse effects. The third dataset is a collection of NSF funding abstracts in various research disciplines [26]. The fourth dataset is the famous collection of Reuters news articles [28]. The fifth dataset is a set of tweets related to Iran's presidential election [27]. The sixth dataset is the *18th Century Collections Online*, a set of transcribed texts from publications in the 18th century, accessible through University of Michigan Library under the name ECCO-TCP [29]. The seventh dataset is a collection of children's fiction assembled by Alan Liu [30]. Tables 4.1 and 4.2 list some general statistics on these datasets.

The definition of "a document" varies from one dataset to another and is ultimately the user's choice. In the online forums datasets, for instance, we have taken each thread as one document. One could also regard every single post in a thread as an individual document. Similarly, for datasets of articles, such as the NSF abstracts datasets or the Reuters news articles, we use the entire article as one document but a smaller piece like a paragraph could also be used. In the 18th Century Collections we take a paragraph as a document and in the Children's Fiction dataset we use each sentence as one document.

	Mothering Threads	Cafemom Threads	NSF Abstracts	Iran Tweets
Number of Documents	26,942	139,455	132,371	2,665,947
Size of Vocabulary	7524	24,389	11,829	11,304
Number of Tokens	25,983,317	521,888,703	15,267,732	19,693,219

Table 4.1: Datasets Statistics: Mothering.com Threads, Cafemom Threads, NSF Abstracts, Iran Election Tweets.

	Reuters News	18th Century Collections	Children's Fiction
Number of Documents	297,141	88,929	761,133
Size of Vocabulary	22,141	26,786	5,781
Number of Tokens	36,363,812	95,265,402	10,470,989

Table 4.2: Datasets Statistics: Reuters News Articles, 18th Century Collections Online (ECCO-TCP), Children's Fiction.

4.2 Evaluation of Summary Topics

Our compression of the document-level information about word combinations that appear in the corpus happens in two stages. First, creating the co-occurrence network, that means we only keep pair-wise co-occurrence of words, from which we would like to infer grouped co-occurrence patterns. Second, regarding bottom-layer clusters as representative of a document group, which further reduces our information into n_c context PageRank profiles. n_c is the number of bottom-layer clusters or contexts.

For some perspective about these numbers, the online forums dataset from Mothering.com has 26,942 documents and nearly 26 million tokens (words in total). For this dataset we extract only $n_c = 440$ bottom-layer clusters further used to build the context-term matrix that contains context profiles over 7524 words of the vocabulary.

As a means of showing topic information is preserved after these compressions, we compare our summary topics extracted by factorizing the context-term matrix, to LDA topics extracted from LDA's generative model which seeks to statistically describe the observed documents. LDA is a well-established method of summarizing a document set, whose results have been the subject of many studies [31] [32] [33]. Therefore, if the summarization method described in section 3.6.1 finds topics that are similar to the topics LDA extracts, it suggests our set of PageRank profiles contain the information necessary to find the most significant topics of the dataset.

The sparse regression method we have used to map topics can also be used to investigate if an LDA topic has any equivalents in our topic.

For an LDA topic i's distribution over the vocabulary stored in row $\beta_{i\star}$ of the β matrix, we find if it can be reconstructed by a linear combination of our summary NMF normalized topic vectors given for a topic j by:

$$\hat{h}_j = \frac{H_{j\star}}{\|H_{j\star}\|_1} \tag{4.1}$$

Because of our topics' contextual focus, we use the z-score vector of LDA topics β_i^Z which also boosts contextually important words compared to the actual probability values in β_i. The z-score

for entry β_{it} using a binomial distribution null hypothesis for word appearances is calculated as [1]:

$$\beta_{it}^{Z} = \frac{N_T.p(z=i).\beta_{it} - N_T.p(z=i).p_t}{\sqrt{N_T.p(z=i).p_t(1-p_t)}} \tag{4.2}$$

Where N_T is the total number of tokens in the corpus, $p(z=i)$ is the probability topic i will be chosen for a single token, and p_t is the global probability estimate of word t given as:

$$p_t = \frac{\text{Total number of appearances of word } t \text{ in corpus}}{N_T} \tag{4.3}$$

Now with the z-score values calculated, we find the equivalent of a normalized LDA z-score vector $\hat{\beta}_{i*}^{Z}$ in our collection of $\hat{h}_j$ vectors which are factorization results of the context-term matrix:

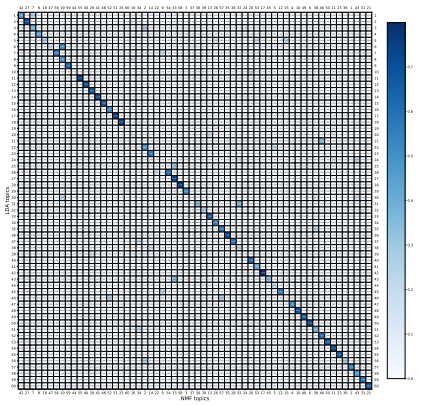
$$\hat{\beta}_{i}^{Z} = \frac{\beta_{i\star}^{Z}}{\|\beta_{i\star}^{Z}\|_1} \tag{4.4}$$

$$\hat{\beta}_{i}^{Z} \approx \sum_{j=1}^{k} \alpha_j \hat{h}_j \tag{4.5}$$

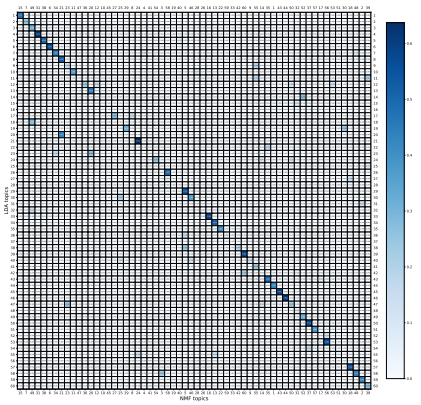
Which as before is solved by minimizing the mean squared reconstruction error and an $L1$-norm penalty term on coefficient α. A heatmap of the solved coefficients for every LDA topic is displayed in figure 4.1 for similar k topics from both LDA and summarizing NMF approaches. Most LDA topics are mapped with a high coefficient to one of our summary topics, and some are divided into different topics.

Both LDA and the factorization procedure are non-convex optimization problems prone to reaching local minima, which means their results will be different from one run to the next, depending on the starting seed of the algorithm. To minimize the effects of such randomness in checking if an LDA topic has an equivalent in our summary topics, in the next step we run the summarization NMF algorithm four times and take the maximum of the LDA topic's mapping coefficients from each run, then the maximum of these four values is selected as the equivalent of the LDA topic. The results of this selection can be seen in figure 4.2 which shows more LDA topics than before will have a close relative in one of the different runs of the summarization algorithm.

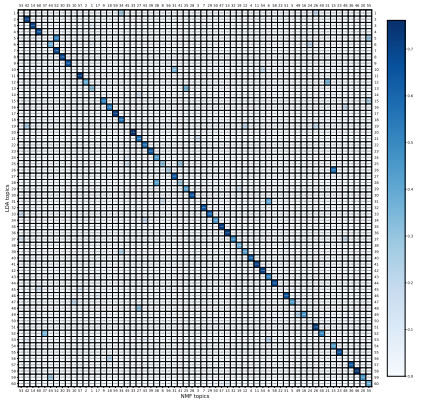
Another approach for comparing the two different outcomes is to check if LDA results for $k = 60$ topics show up in summary NMF results for $k = 100$ topics. Such a mapping is shown for the Mothering.com dataset in figure 4.3 as an example. One can see after appropriate re-ordering



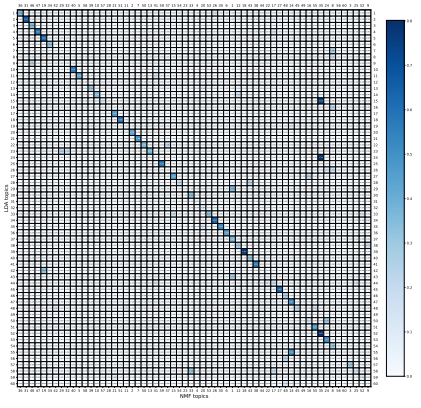
(a) Mothering Threads, $k = 60$ in both methods



(b) Cafemom Threads, $k = 60$ in both methods



(c) NSF Abstracts, $k = 60$ in both methods



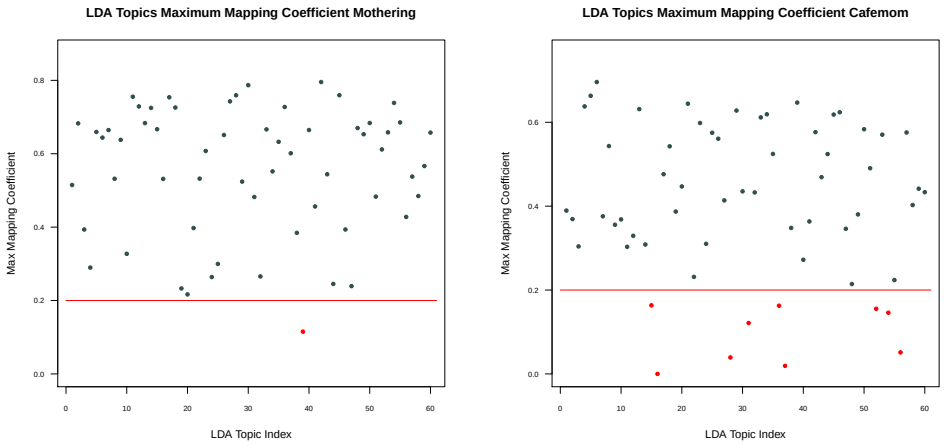
(d) Iran Election Tweets, $k = 60$ in both methods

Figure 4.1: Mapping coefficients from LDA topics and summary NMF topics for four datasets.

of topics, the main diagonal of this mapping has more "filled" entries on the main diagonal than the result of similar mapping shown in figure 4.1(a).

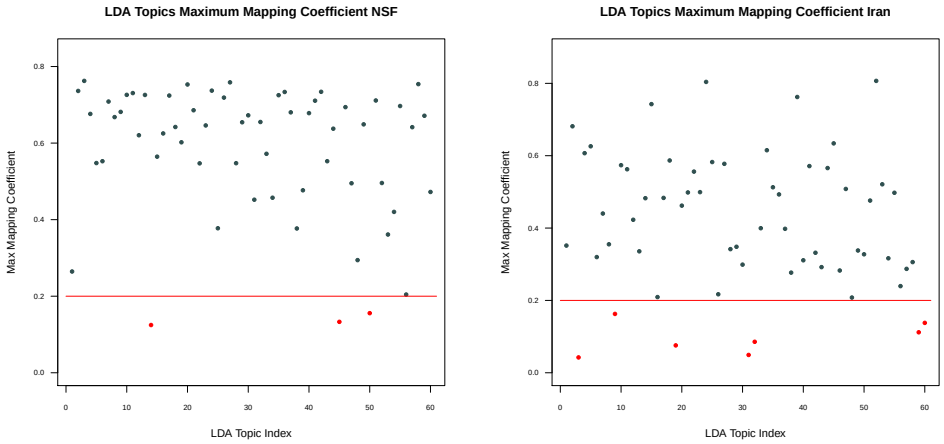
For instance, an LDA topic in Cafemom about "smoking" with the following top 10 words is not matched when $k = 60$ summary topics are extracted:

- smoke smoking striving alcohol sympathetic tender tolerant compassionate aged marijuana



(a) Mothering Threads

(b) Cafemom Threads



(c) NSF Abstracts

(d) Iran Election Tweets

Figure 4.2: Maximum mapping coefficient from each LDA topic to summary NMF results of four different runs on each of the four datasets. For all of the displayed plots $k = 60$ in both methods.

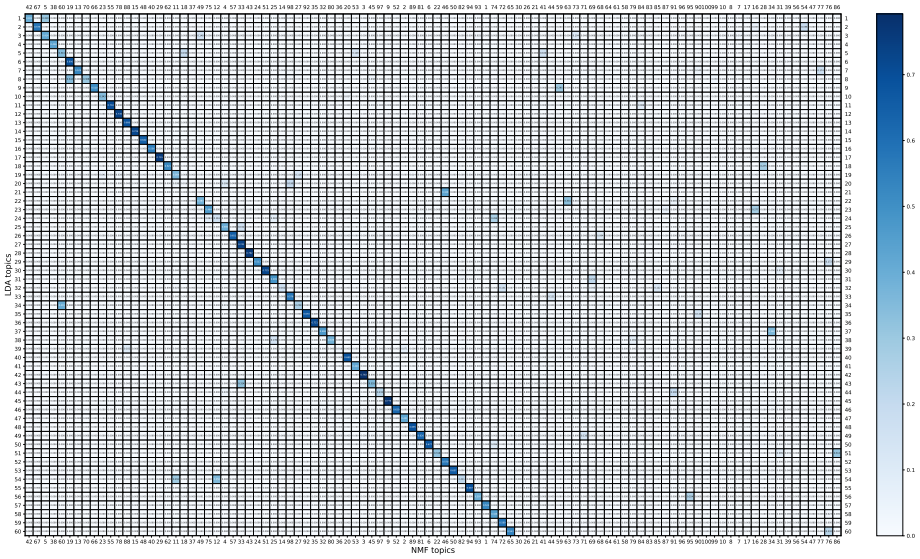


Figure 4.3: Mapping coefficients from LDA with $k = 60$ topics to summary NMF with $k = 100$ topics for Mothering.com threads.

When we extract $k = 100$ summary topics, however, a matching topic is found with a mapping coefficient of 0.42 and the following top 10 words:

- alcohol smoking drugs smoke drinking drug addiction drink illegal cigarettes

By putting a chosen threshold on the maximum mapping coefficient of an LDA topic, we can decide if the LDA topic is present in our set of NMF topics and then find the number of LDA topics with a match in our results. In these plots a threshold of 0.2 is chosen, so LDA topics that map to a summary NMF topic with a mapping coefficient larger than 0.2 are determined as matched. Note that the z-score vector of LDA topics that we are mapping to summary NMF topic vectors —components of the context-term matrix— are inherently different in their distribution of values. Therefore, a mapping coefficient of even $0.3 \sim 0.4$ usually signals a good conceptual overlap when the actual word combinations of topics are judged. We can then increase k for summary NMF and identify if more LDA topics find matches. Table 4.3 summarizes number of matched $k = 60$ LDA topics as we increase k for summary NMF.

	Covered k=60 LDA topics for different k settings in summary NMF		
	$k = 60$	$k = 80$	$k = 100$
Mothering.com Threads	59	59	60
Cafemom Threads	51	54	55
NSF Abstracts	57	59	58
Iran Election Tweets	54	56	56
Children's Fiction	51	52	55
18th Century Collections	48	49	52
Reuters News Articles	53	60	59

Table 4.3: Number of LDA topics covered in various k summary NMF topics. For all datasets $k = 60$ LDA topics are analyzed. In some cases, number of covered topics stays constant or even decreases when going from $k = 80$ to $k = 100$ summary NMF topics. This is because for larger numbers topics are divided differently and thus some old topics are distributed across several new topics. By taking the number of LDA topics covered in at least one of these three different NMF summaries of each dataset, on average 94% of a dataset's $k = 60$ LDA topics are covered in the summary.

The set of summary topics extracted by our summary NMF method create a division of topics that would not be exactly similar to what LDA extracts. Some of these slight differences arise from the different mindset that goes into our contextual analysis of word co-occurrence patterns. For instance, on the Mothering.com using LDA with $k = 60$ topics, we observe a topic on "clinical tests" with top 10 words:

- test tb blood tests testing titer titers ppd positive tested

Whereas in our set of summary topics the "clinical tests" concept is distributed across different contexts it appears in, such as "pregnancy", "immigration", and "STDs". The top 30 words of these summary topics are listed below (the numbers in parentheses denote the rank of a word that appears later in the topic):

- pregnant pregnancy women woman ob pg birth rubella fetus pregnancies baby trimester unborn ***test*** born delivery mother ttc ***blood*** midwife prenatal mothers labor babies ***tested*** postpartum antibodies ***titers*** immunity future
- ***tb*** immigration visa applying immigrants bcg citizen x-ray xray employer immigrant employment fee clinicals citizens hire ***ppd*** hired semester employee sputum application apply college classes tuberculin rn pay students comply
- sex cancer hpv girls sexually women gardasil transmitted cervical std sexual pap active genital cancers woman routine warts strains infected smears partners condoms stds tested paps screened merck men female ... ***testing(38) tests(45) test(49)***

In the probabilistic generative model of LDA, it makes sense to create a topic solely on different clinical tests, which will be partially present in documents (or contexts) that are mainly about pregnancy or immigration. In contrast, our contextual factorization approach puts much higher emphasis on simultaneous presence of the group in documents (or contexts). Therefore, a word like "tb", which is short for "tuberculosis" and has an associated skin test performed for immigration purposes, is very unlikely to be combined in the same topic with "blood" and "titers", which are

here mostly mentioned for blood tests during pregnancy. This LDA topic is thus distributed among its contextually relevant topics, simply because this combination does not occur in documents.

The word composition quality of summary topics can also be measured by comparing the top LDA words to top words of their best-match summary topic from the selection method described above. For an LDA topic we find how well its top words are covered by our summary topic. Word coverage plots can be drawn by a curve on the following points: for an index r_l what is the smallest index r_n that at least 80% of the top r_l words of the LDA topic are covered in the top r_n words of its best-match summary NMF topic. The percentile condition is used so that single outliers do not greatly disturb the coverage curve. These word coverage plots are shown for a sample set of LDA topics in figure 4.4 .

Another measure to judge how well the top LDA words are covered is the *z-score-weighted coverage* of top words. More specifically, one can find how many of the top 100 words of an LDA topic are also in the top 100 words of the summary NMF topic, with weights to account for their importance. The coverage score c_{ij} from LDA topic i to summary NMF topic j is calculated as:

$$c_{ij} = \frac{\sum_{r=1}^{100} \beta_{ir}^{Z} \delta_{ij}^{N}(r)}{\sum_{r=1}^{100} \beta_{ir}^{Z}} \tag{4.6}$$

$$\delta_{ij}^{N}(r) = \begin{cases} 1 & r^{th} \text{ word of LDA topic } i \text{ is in top 100 words of summary NMF topic } j \\ 0 & \text{otherwise} \end{cases}$$

These values are plotted in a heatmap in figure 4.5 and show most of the important LDA topic words are covered in the relevant summary topic.

4.3 Evaluation of Layered Clusters

We make use of the notion that bottom-layer clusters correspond to groups of similar documents. The first implied claim to be tested here is that documents are in fact built on bottom-layer clusters which set a context; they cannot be built on middle-layer concepts without any use of bottom clusters. This can be shown by examining if all documents have words from bottom-layer clusters in them. For a document d we can find its share of bottom clusters by finding $\frac{\sum_{C_i} N_{d,C_i}}{N_d}$, where N_{d,C_i}

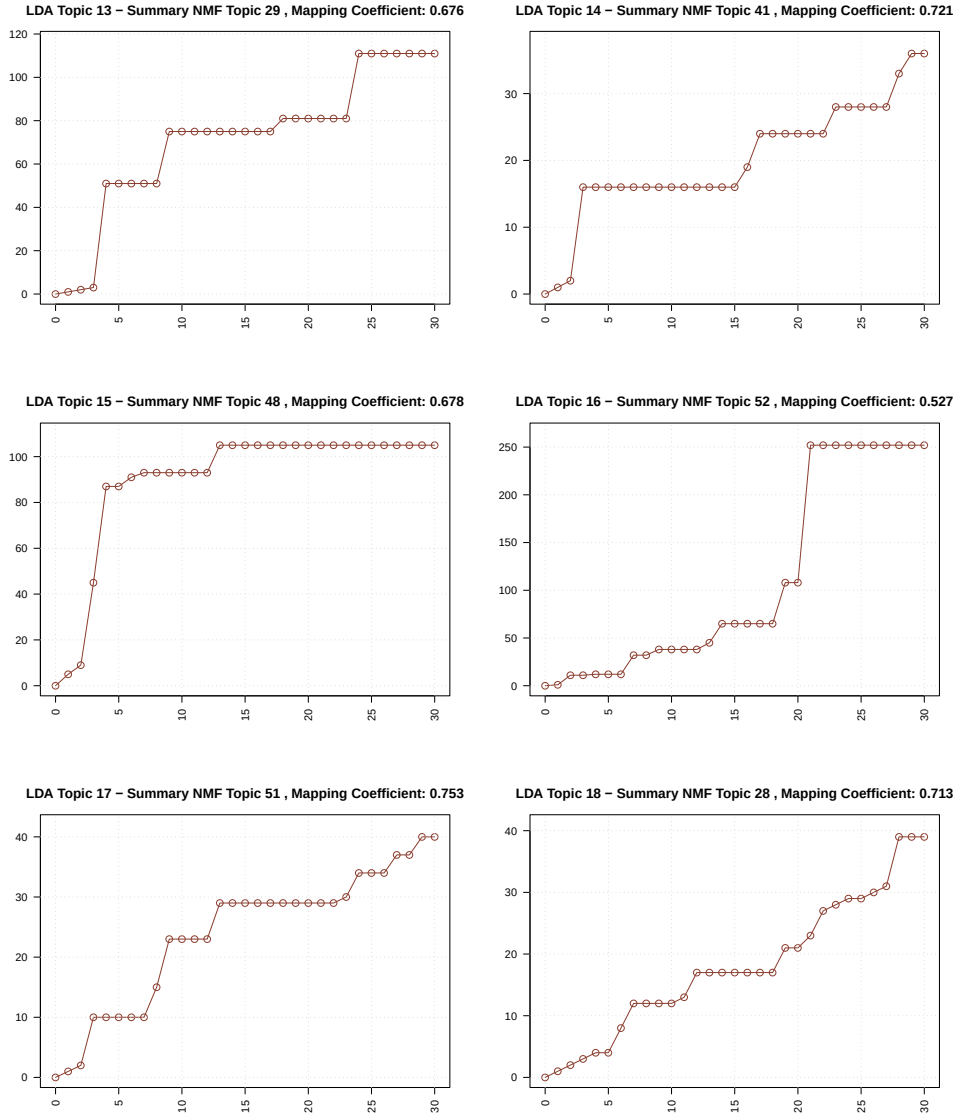
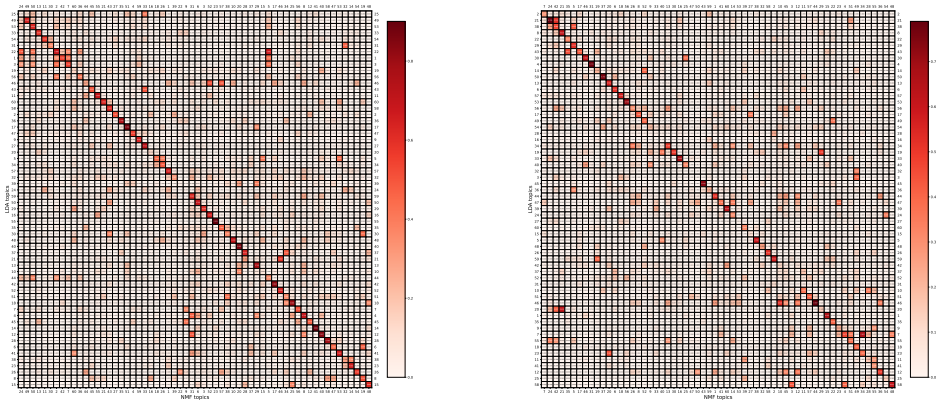
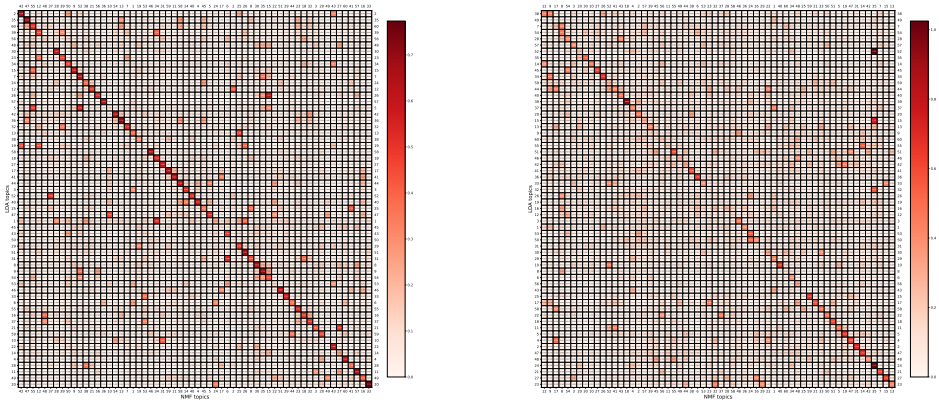


Figure 4.4: Word coverage curves for top words of 6 Mothering.com LDA topics in their best-match summary NMF topic. Each point (i, j) drawn on a plot displays the smallest j for which 80% of top i words of the LDA topic are covered in the top j words of the summary NMF topic.



(a) Mothering Threads, $k = 60$ in both methods (b) Cafemom Threads, $k = 60$ in both methods



(c) NSF Abstracts, $k = 60$ in both methods (d) Iran Election Tweets, $k = 60$ in both methods

Figure 4.5: Z-score-weighted coverage of top 100 LDA topic words in top 100 words of NMF topics. Topic numbers are permuted to reduce the matrix bandwidth such that higher values tend to get closer the main diagonal [41].

is the number of words in document d that are in bottom cluster C_i and N_d is the total number of words in the document, so we are calculating the fraction of document words that are from bottom clusters. Figure 4.6 shows a histogram of these values over all documents for two of our datasets as examples. As observed in the plots, there are very few documents with no share from the relatively rare words of bottom-layer clusters in them. For both datasets, an average document has about 0.1 of its words from bottom-layer clusters.

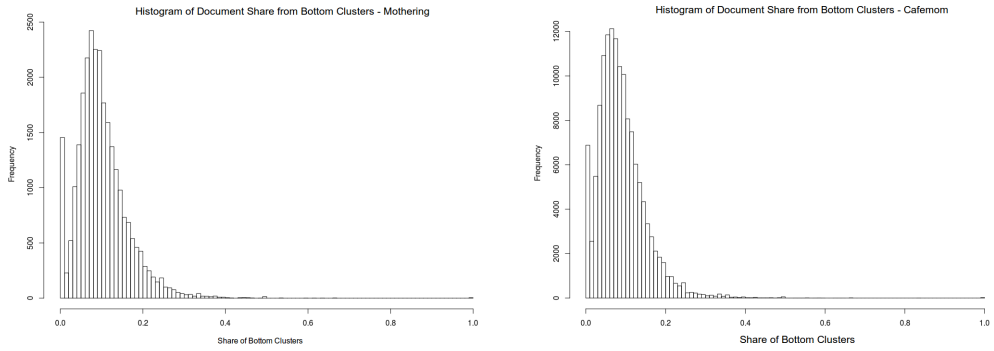


Figure 4.6: Histogram of documents' fraction of words from bottom-layer clusters for Mothering (left) and Cafemom (right).

We also need to test validity of the higher layer clusters. Our layered clustering is based on contextual co-occurrence of words. This means our criteria for putting some word groups of a certain layer L together in one cluster is their simultaneous presence in contexts defined by core clusters which are representative of document groups.

We evaluate our middle layer clusters by checking if they show high modularity in the co-occurrence network. The modularity measure put forth by Newman [43] is a way of examining quality of a given clustering on a network. Modularity of a cluster is calculated by comparing observed edges between pairs of nodes in the cluster to the expected edges if node degrees were kept intact but edges were made at random. For an undirected unweighted network with adjacency matrix A, a cluster's modularity is given by [44]:

$$\mathcal{Q} = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) \tag{4.7}$$

Where $k_i = \sum_t A_{it}$ is the degree of node i, m is the total number of edges, and s_i is the binary cluster membership indicator with $s_i = 1$ if node i is inside the given cluster and $s_i = -1$ otherwise. The average value of modularity is very close to zero for randomly selected clusters, and positive for a relatively dense cluster. The idea can be straightforwardly extended to weighted networks by using the weighted adjacency matrix instead of the binary matrix A [45]. It can also be modified to evaluate a given clustering of the network into multiple communities as again mentioned in [44], but since we are not doing a hard clustering in our layers, we use the single-community version to assess the quality of each cluster individually.

To evaluate each layer cluster, we take the subgraph of the co-occurrence network formed by the cluster's top 10 words and calculate the subgraph's modularity by inserting in the above equation $s_i = 1$ if i is one of these 10 nodes and $s_i = -1$ otherwise. We then compare this value against the standard deviation of modularities given by random subgraphs created by 10 random nodes in the same layer L. Figure 4.7 shows modularity values for layer clusters of a few different layers of Mothering.com. A plot of the average modularity of layer clusters at each layer versus the average modularity for random clusters of the same layer is shown in figures 4.8 and 4.9. It is worth noting that in the Iran election tweets dataset and the NSF abstracts dataset, the top layers have very few words and this kind of analysis based on top 10 cluster words cannot be carried out. These very few words in top layers, such as "research", "project", and "study" in the NSF abstracts dataset, might as well be put in one single layer cluster that is used by most contexts.

Despite loss of contextual information in higher layers of the co-occurrence network as described in section 3.5 —which would partly suppress modularity of contextually significant groups of words— the modularity of layer clusters are consistently high. High modularity values of these higher layer clusters supports two arguments:

1. These higher layer clusters extracted from our context PageRank profiles are in fact co-occurring groups of words in documents, so the layer clusters are meaningful.

2. The bottom-layer clusters represent groups of similar documents. The higher layer words that are strongly connected to a bottom-layer cluster of co-occurring words tend to also co-occur in documents themselves. This means documents that involve the bottom-layer

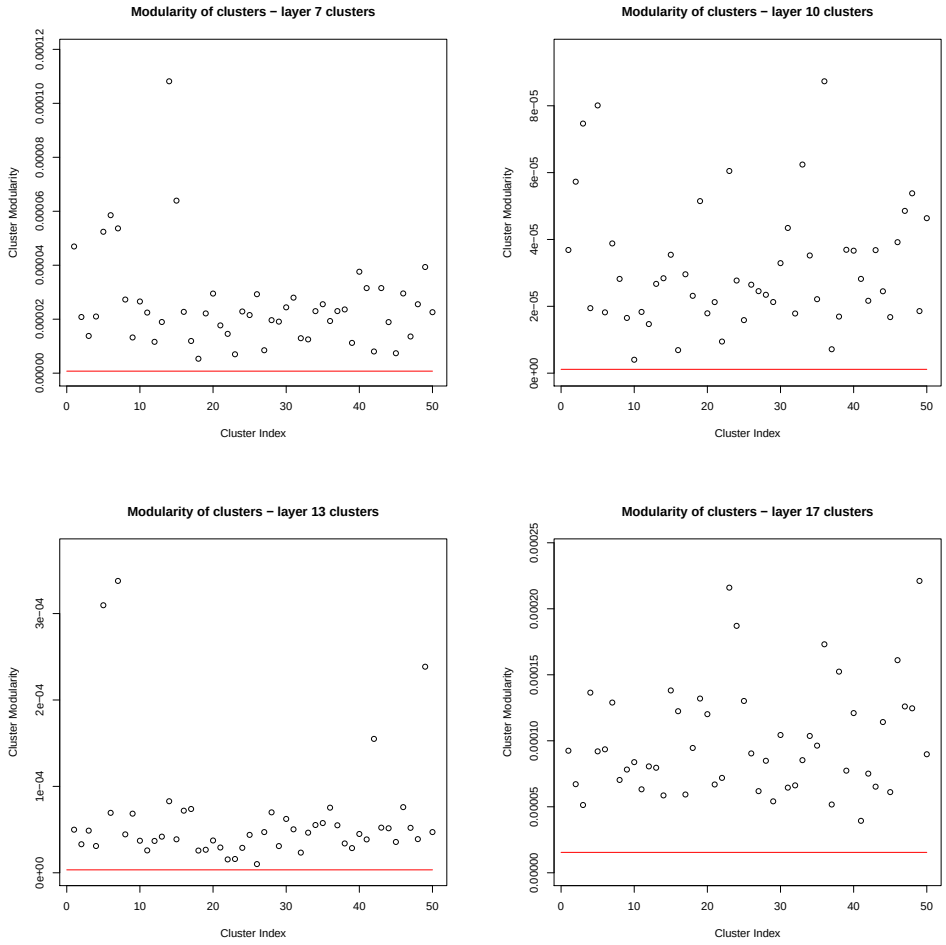


Figure 4.7: Modularity of layer clusters for layers 7, 10, 13, and 17 on Mothering.com inside the layer's subgraph of co-occurrence network. The red line indicates standard deviation of modularity for 10 random nodes in the layer. The average value of modularity for randomly selected clusters is relatively very small compared to the standard deviation.

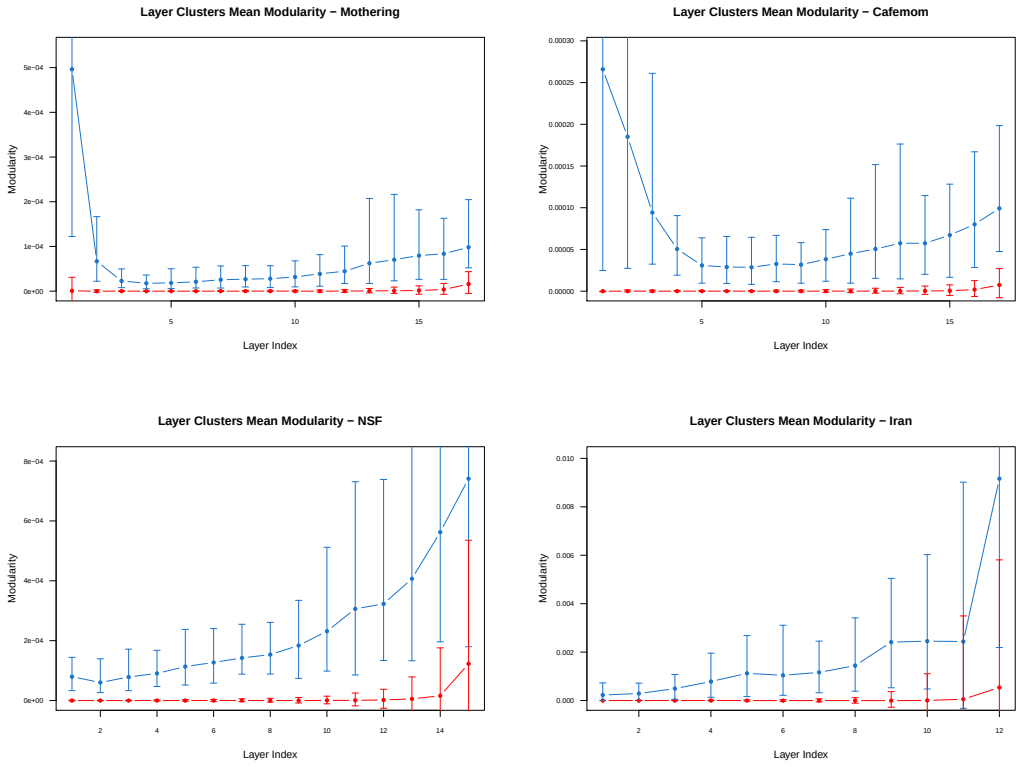


Figure 4.8: Average modularity of layer clusters (blue dots) and random clusters of same size in the same layer (red dots). Boxes on points display 90% confidence intervals. These values are shown for four datasets of Mothering.com threads (top left), Cafemom threads (top right), Iran Election Tweets (bottom left), NSF abstracts (bottom right).

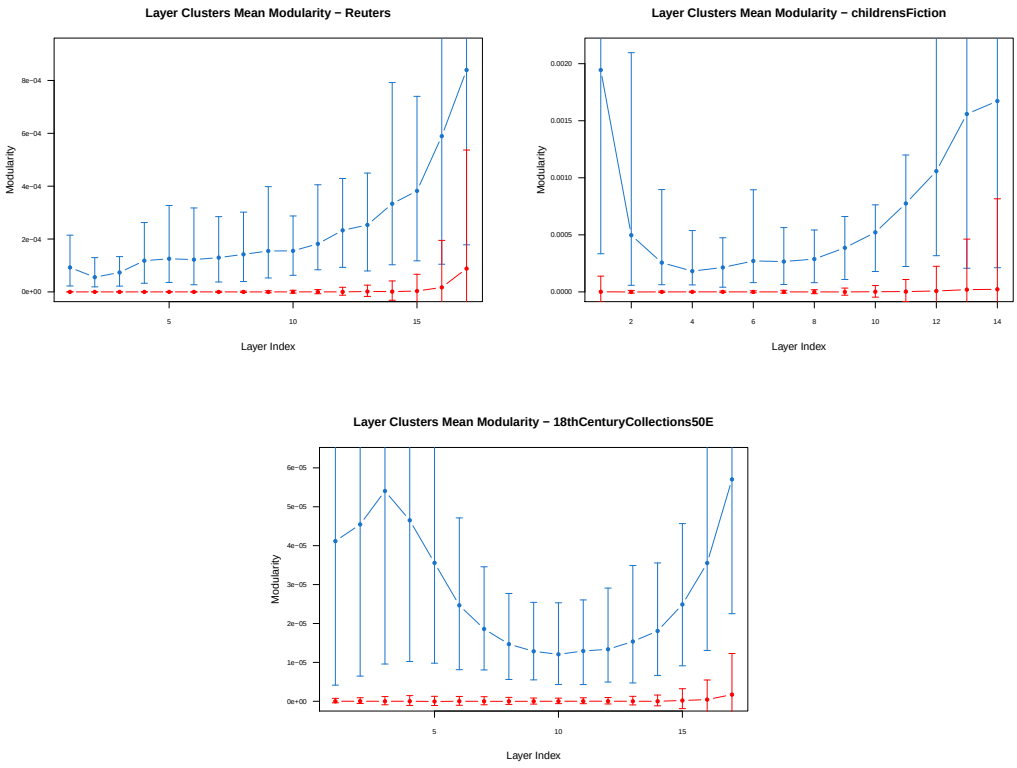


Figure 4.9: Average modularity plots of layer clusters described in figure 4.8, for three datasets of Reuters news articles (top left), Children's Fiction (top right), and 18th Century Collections (bottom).

cluster words also include similar words as each other in higher layers and so are similar throughout the PageRank spectrum (These documents are the source of co-occurrence in words of associated higher layer clusters). A more direct assessment of this notion can be performed by mapping individual documents back onto the bottom clusters and comparing their word distributions.

CHAPTER 5

Discussion and Further Work

5.1 Discovering Structures from Data

The extracted hierarchical structure of concepts can be a powerful tool to gain insights about a text dataset. To demonstrate one such instance, we look into the case of vaccination discussion forums on Mothering.com and Cafemom. These datasets provide a great example of benefits of extracting concept structures in application. Let us start by giving a short background on the story.

Recent outbreaks of measles, pertussis and other vaccine preventable diseases (VPDs) in the United States have emerged as a serious public health crisis. These outbreaks have been attributed to the increasing number of under- and unvaccinated children in various communities [51] [52]. While it is broadly understood that this critical level of under vaccination can be attributed to the exemption seeking behavior of parents, it is not entirely clear what is driving this behavior [53]. Social media sites dedicated to parenting discussions likely play a role, with the frequent exchange of persuasive personal opinions contributing to an accepted objective of exemption-seeking in the community.

As discussed in [4], the resonant narrative in these anti-vaccination discussions is that vaccines pose the real threat, not VPDs, and exemptions emanate as a strategy to avoid them. These strategies can be observed as a part of the concept hierarchy extracted from Mothering.com. This piece that is displayed in figure 5.1 summarizes the exemption strategies and mechanisms and how they all relate in the larger context of school enrollment.

Cafemom discussions are less biased than the well-known anti-vaccination theme of Mothering.com. Nevertheless, smaller structures on exemption requirements can be observed from this

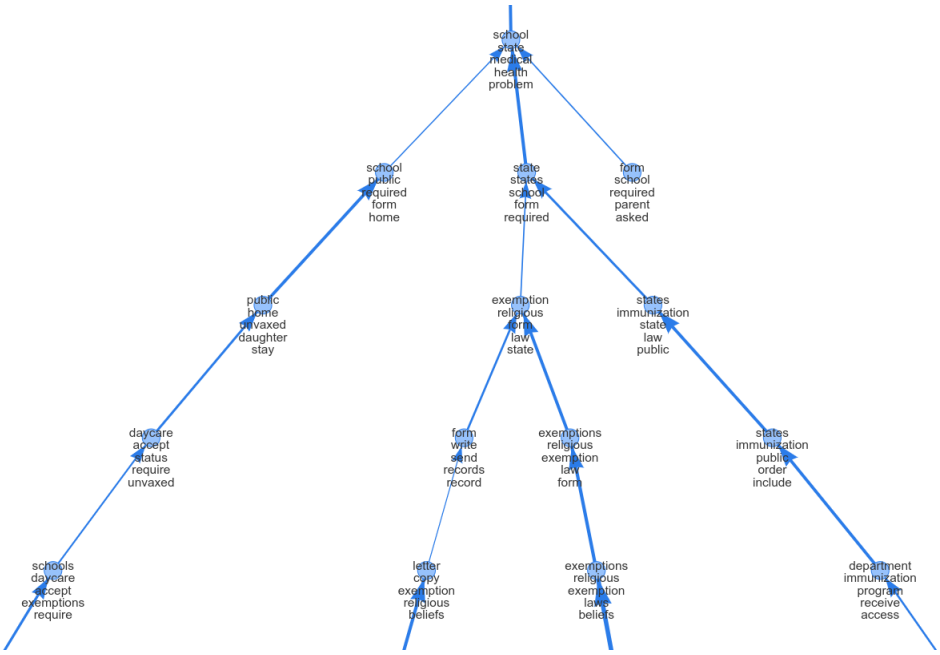


Figure 5.1: A crop of the extracted hierarchical structure of Mothering.com that relates to "exemption". One can see different branches of concepts about "religious exemption" , "required forms", "state laws", and "school regulations" merging at the higher layers to indicate their shared scope of getting exemption from vaccines to enroll children in school.

dataset as well as shown in figure 5.2 . The higher diversity of views on Cafemom can be seen in the discussions on vaccine safety. While Mothering.com conversations imply a strong link between vaccine ingredients and autism as previously discussed in figure 3.17, these links are weaker in Cafemom threads and mostly in the form of discussing vaccine safety research, such as the example in figure 5.3.

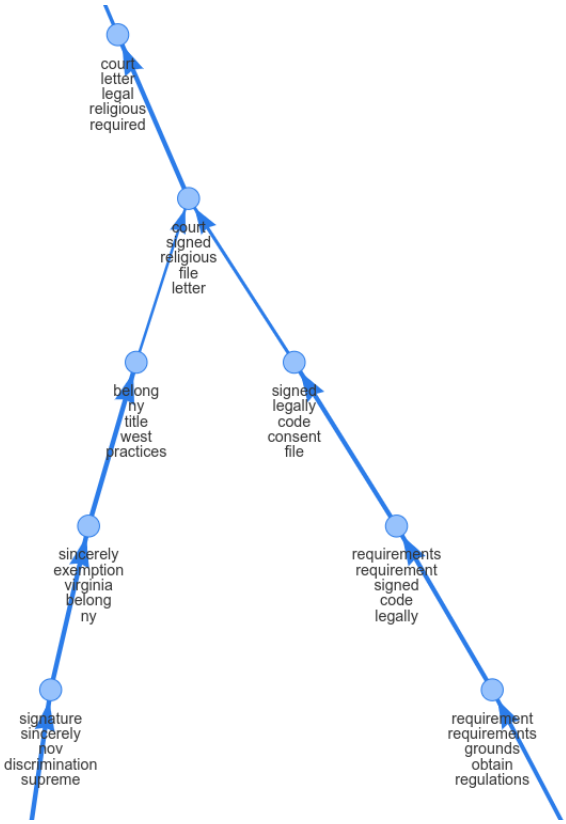


Figure 5.2: A crop of the extracted hierarchical structure of Cafemom showing chains related to "exemption requirements" and "state laws".

The shape of the hierarchical structure depends very much on the dataset. Although there is an overall tendency for clusters to merge when going towards the *more general* higher layers — and it is more so for topically focused datasets such as Mothering.com threads about vaccination or NSF abstracts containing research materials— in some cases the structure is mostly flat, with many divided chains that independently go all the way from bottom layer to the top. One such case is the dataset of 18th Century Collections whose hierarchical structure can be seen in figure 5.4 . This dataset contains various types of text published in the 18th century and the collection does

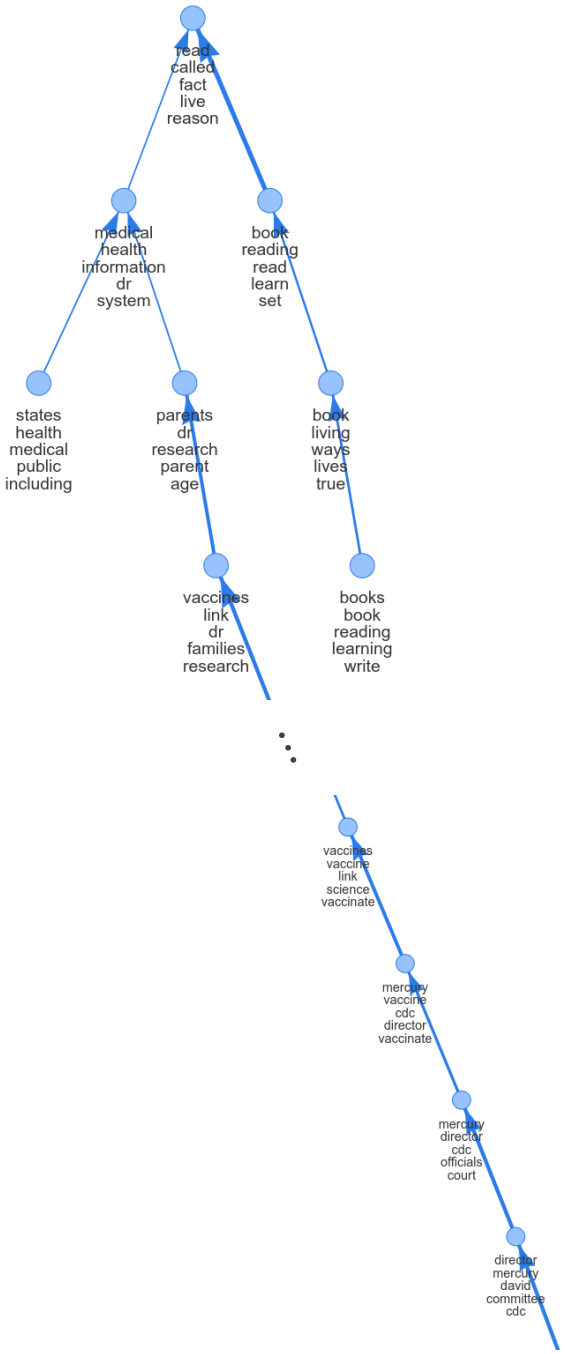


Figure 5.3: A close-up of the hierarchical structure of Cafemom concepts relating to "vaccine safety". The bottom plot shows a related chain in the middle layer and the top plot shows continuation of the same chain at the top layers that mixes with concepts on "reading and research about vaccines".

not have a specific scope. These texts naturally have very divided topic structure and much fewer connections across different hierarchical chains, as evident in the distant plot. A close-up of two chains in the hierarchy is shown in figure 5.5 .

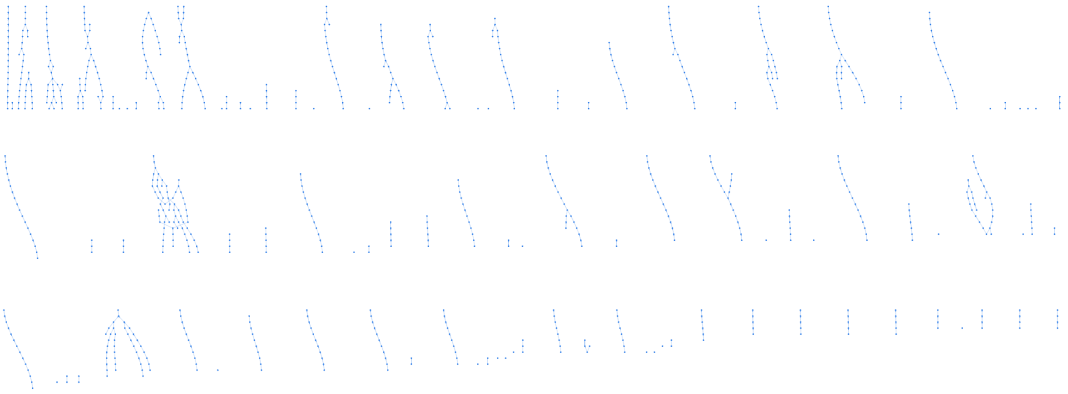


Figure 5.4: A distant view of the hierarchical structure extracted from 18th Century Collections. The plot is broken into three parts due to size; the 3 figures from top to bottom display the left, middle, and right parts of the hierarchy. The large variety in this collection and lack of a central scope, creates a very divided hierarchy of mostly independent chains.

5.2 Actant Clustering

Another application of our contextual analysis is in actant-interaction models such as [4]. These models provide a method for story detection and aggregation on social media to identify key narratives. The story is presented in the form of a network of actants as nodes and their pairwise interactions in a particular *context* as edges. The term "context" is overloaded here and for the story model refers to a setting that implies a particular set of actions between actants.

An abstract form of the story model from [4] is shown in figure 5.6. For a story revolving around a set of actants $A_1, ..., A_n$, the narratives can be divided into a set of story contexts. In each story context, the story is summarized as a set of interactions (relationships) between actants as shown in the figure. Therefore, an edge between actants A_1 and A_2, for example, carries a set of relationships $\mathcal{R}_{12} = \{R^1_{12}, R^2_{12}, \cdots, R^r_{12}\}$ that exist between the two actants, and the significance of each relationship in this story context.

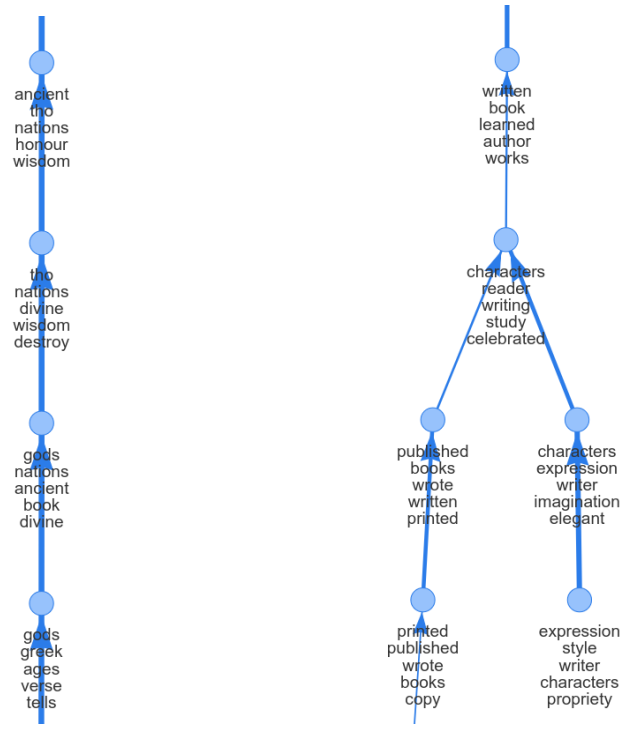


Figure 5.5: A close-up of two hierarchical structure pieces from the 18th Century Collections dataset. These pieces are related to "greek mythology" (left) and "writing" (right).

In the set of discussions around vaccination, some main actants of the story include parents, children, doctors, schools, government, vaccines, VPDs (Vaccine-Preventable Diseases), and exemptions. Pairwise relationships occur in the form of (actant, relationship, actant) tuples, for instance (doctors, recommend, vaccines), (parents, protect, children), (children, contract, VPDs), (schools, require, vaccines), and (parents, seek, exemption). Significant relationships are inferred by finding recurring relationships in sentences throughout the dataset. The following sentence for example:

```
State laws may require vaccination for children, even if
they are not attending public schools.
```

Implies the relationship (government, requires, vaccines). By parsing the sentences and finding relevant interaction structures, like a (subject, verb, object) tuple for instance, one can directly extract the relationship (state, require, vaccination) from this sentence. In order to aggregate all

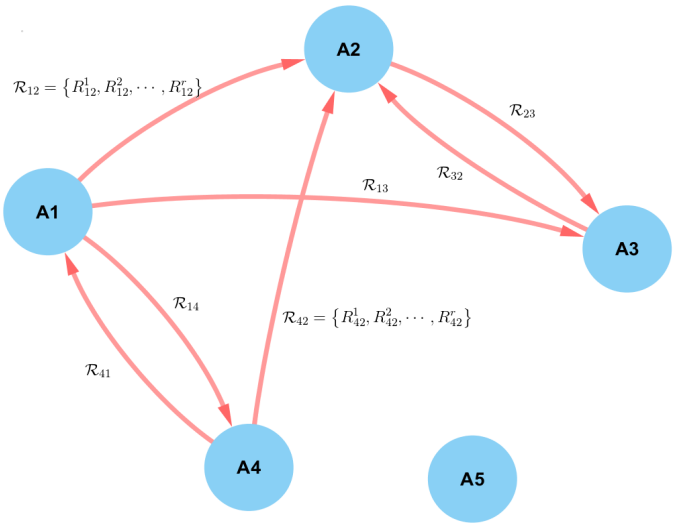


Figure 5.6: Contextual Network Model of Stories. Nodes represent actants $A_1, ..., A_n$, and edges carry information about relationships and interactions that arise in a particular story context between pairs of actants.

relationship instances, we need to find terms in which an actant is realized. For example, in the context of vaccination discussions, "state", "CDC", "the feds", and "officials" all refer to the same actant under the umbrella term "government".

One strategy to find such groups of terms is the use of word embeddings that map words onto points in a k-dimensional space. On this space, words with semantic similarities will be mapped to relatively close points [54]. One can then find clusters of close words to find semantically similar nouns that likely refer to the same entity/actant.

In our framework, the contextual features extracted for words as a by-product of our analysis of context profiles can be used as word embeddings. These feature vectors are stored in the columns of our context-term matrix T. These features are, however, conceptually different from outputs of algorithms such as the popular word2vec [54] that create embeddings from sentence-level semantic roles of words. In this case, contextual activity of words drive their embeddings. In other words, a word's feature vector contains the document-level contexts it participates in.

These context-level word embedding vectors can also be used to find groups of words with

similar context activity that likely refer to the same actant. A simple solution is to threshold these term-context values and create a binary vector for each word that contains 1 for contexts the word appears in and 0 elsewhere. A context co-occurrence matrix can then be formed on recurring nouns of our dataset and finding dense communities on this network can give us the target groups. Two examples of these noun clusters are listed in table 5.1.(a).

Cluster 1	relationship weight doctors listens milestones wbvs pediatrician respects checks chiropractors doc wbv visit retract visits pedi trail
Cluster 2	cytokine bile models pathogenic ... contrast colonization polio experts ... infection poultry cases....

(a) Sample clusters extracted directly from context co-occurrence network of nouns.

Cluster 1	**doc(0.544) pediatrician(0.461) doctors(0.400) pedi(0.332)** visit(0.095) visits(0.0911) chiropractors(0.015) respects(0) relationship(0) weight(0) trail(0)
Cluster 2	**polio(4.01) infection(3.62) disease(3.32) dis(3.16) flu(3.00)** adults(2.92) cases(2.80) ...

(b) Top words of each cluster based on weighted node degree in the relationship similarity network of cluster words

Table 5.1: Actant name clusters extracted from context co-occurrence of recurring nouns in Mothering.com. Cluster 1 contains different names that refer to "doctors" and cluster 2 mainly contains "diseases". The second example cluster is quite large so only parts of it are shown here.

These clusters provide groups of contextually similar words. In addition to similarity in contextual activity, different names that refer to the same actant must also engage in similar relationships. For examples, we will have instances of both (doc, recommends, vaccine) and (pediatrician, recommends, vaccine). By adding this constraint and filtering the cluster to only contain words with similar relationship pattern, we get main words of the cluster that refer to an actant. A quick real-

ization of this constraint is done by forming a small network of a cluster's words as nodes, where edge weights between two words (i, j) is the Jaccard index of their relationships $\frac{|R_i \cap R_j|}{|R_i \cup R_j|}$. Getting top words of this small network based on some centrality measure such as node degree or PageRank will now give us main words of the cluster. The result of this filtering approach applied on the example clusters is shown in table 5.1.(b) that shows top cluster words referring to one actant.

The results in table 5.1 stand as a proof of concept how grouping of different names for an actant can be done in this framework. A more elegant joint optimization problem to find actant name clusters may also be devised that simultaneously minimizes:

1. Variation of context feature vectors among words of a cluster.

2. Variety of relationships patterns the cluster words engage in.

This application of the embedded word-contexts feature vectors is to be further investigated in future works.

CHAPTER 6

Tools and Other Analyses

6.1 Community Detection

With the modified random walk model for document generation which includes a reflection probability from high-PageRank nodes, we can extract communities in the graph that represent "contexts" using any random-walk-based community detection algorithm. One such method is the walk-trap algorithm proposed by Pons et al. [34]. The walk-trap algorithm defines a distance between pairs of vertices using the probability of going from one to another in a fixed number of steps. A K-median clustering is then used to find communities using these distances. Another more recent community detection algorithm based on random walks is the InfoMap algorithm by Rosvall et al. [35]. The InfoMap algorithm finds communities through minimization of description length of a random walker's movements with a two-level coding scheme. The grouping of nodes in the higher level will correspond to dense communities. InfoMap is a better match with our framework, as contexts are seen as abstract higher-level dynamics controlling the random walker.

6.1.1 InfoMap

In InfoMap, given a partitioning into communities, random walks are coded with a two-level coding scheme that assigns an enter code and an exit code to each communities, and as long as the walker stays within a community it has entered, its position is coded by a dictionary of codewords specific to nodes inside the community. Therefore, two nodes in two different communities can have similar codewords, since the enter and exit codes make clear which community the walker is in, and thus the position is known.

The average number of bits needed to describe a random step is bounded below by the overall weighted entropy according to Shannon's source coding theorem [36]. Community division is therefore chosen to minimize the entropy cost function:

$$L(M) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i) \tag{6.1}$$

The first term accounts for entropy of movements between communities, while the second term sums up weighted entropy values of within-community movements.

$$H(\mathcal{Q}) = \sum_{i+1}^{m} \frac{q_{i\curvearrowright}}{\sum_{j=1}^{m} q_{j\curvearrowright}} \log \left(\frac{q_{i\curvearrowright}}{\sum_{j+1}^{m} q_{j\curvearrowright}} \right) \tag{6.2}$$

$$p^i_{\circlearrowright} = q_{i\curvearrowright} + \sum_{\alpha \in i} p_\alpha \tag{6.3}$$

6.2 Non-Negative Matrix Factorization

Matrix decomposition and low-rank representation is one of the most useful techniques in data sciences to find co-occurrence patterns across observed samples of multivariate data. Non-negative Matrix Factorization (NMF) techniques in particular have risen in popularity over the past decade due to their performance, efficiency, and applicability to a wide range of problems. These algorithms have become prevalent tools in different domains from collaborative filtering [37] to genetics and molecular pattern discovery [39]. A basic version of the NMF problem is posed as follows:

Given matrix $X \in \mathbb{R}^{m \times n}$ with non-negative entries, we want to find non-negative matrices $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$ whose product approximately reconstructs X such that the Frobenius norm of the error $||X - WH||_F^2$ is minimized.

In order to solve this problem, matrices W and H can be initialized with arbitrary values that retain full rank on both matrices, such as values drawn from a uniform distribution. Depending on the cost function that is to be minimized, different update rules can be used to achieve local optima

on W and H. For example, the squared Euclidean distance $\|X - WH\|^2 = \sum_{ij}(X_{ij} - W_{ij}H_{ij})^2$ is non-increasing under the following update rules [38]:

$$H_{a\mu} \leftarrow H_{a\mu}\frac{(W^T X)_{a\mu}}{(W^T WH)_{a\mu}} \tag{6.4}$$

$$W_{ia} \leftarrow W_{ia}\frac{(XH^T)_{ia}}{(WHH^T)_{ia}} \tag{6.5}$$

The immediate benefit of such a framework is its compression of data samples. Assume X is composed of m row vectors of length n containing m observed samples of a multivariate group of n random variables. The factorization above compresses this data into k row vectors of n dimension stored in H, while each of the m data points, i.e. each row of X, is stored in a row of W as a linear combination of these k n-dimensional vectors. In other words, each data point is mapped onto a k-dimensional feature space. In order to create such a compression, the factorization procedure *groups* variables that often rise together in the samples, and assigns to them high values in the same n-dimensional component of H.

As mentioned above, the cost function to minimize in a simple form of NMF is the Frobenius norm of $(X - \tilde{X})$ where $\tilde{X} = WH$ is the reconstructed target matrix. A weighted version of NMF introduces the weight matrix $M \in \mathbb{R}^{m\times n}$ that stores the importance of reconstruction for each entry in X. The updated cost function for weighted NMF now is:

$$F = \sum_{i=1}^{m}\sum_{j=1}^{n}\left\{M_{ij}(X_{ij} - \tilde{X}_{ij})^2\right\} \tag{6.6}$$

which will be equal to zero when $X = \tilde{X}$. The weighted NMF can be solved by simple modifications on the update rules. The new update rules to solve this problem are [40]:

$$H_{a\mu} \leftarrow H_{a\mu}\frac{\sum_i W_{ia}\frac{X_{i\mu}}{M_{i\mu}}}{\sum_i W_{ia}\frac{\tilde{X}_{i\mu}}{M_{i\mu}}} \tag{6.7}$$

$$W_{\nu a} \leftarrow W_{\nu a}\frac{\sum_j \frac{X_{\nu j}}{M_{\nu j}}H_{aj}}{\sum_j \frac{\tilde{X}_{\nu j}}{M_{\nu j}}H_{aj}} \tag{6.8}$$

The weights can be used to take into account "uncertainty of measurement" for each entry in X as proposed in [40]. For our purposes, we use this weight to apply importance of reconstruction on each word as a function of their PageRank. In general, we want more important words, i.e. words with higher PageRank values, to assert higher penalty in the cost function for smaller deviations in reconstruction. Each row in X stores one measurement of intensity of words in a specific context and the columns represent different intensity values for one word over different contexts. Therefore, we use a weight matrix M with column j equal to PageRank of j_{th} word in the vocabulary:

$$M = \begin{bmatrix} PR(1).\mathbf{1} & PR(2).\mathbf{1} & \cdots & PR(n).\mathbf{1} \end{bmatrix} \tag{6.9}$$

The reason for such weighting strategy is two-fold. First, it forces the NMF decomposition to focus on more important words with higher PageRank and by doing so it reserves co-existence of these words and assures coherence in extracted output word combinations. Second, as shown in the context-specific PageRank deviation plots, higher-PageRank words show smaller response to activation from a low-PageRank context and have lower PageRank deviation from their initial state. The weighting matrix can compensate for this behavior and boost their significance.

6.3 Algorithm Complexity

In its original implementation, LDA's time complexity is $\mathcal{O}(MNk)$, where M is the total number of documents, N is the number of words in each document, and k is the number of output topics [3]. This is because the variational inference method, in every optimization step, updates topic assignment multinomial $(\phi_{n1}, \phi_{n2}..., \phi_{nk})$ for all terms n in a document. The LDA model has since been a subject of various studies and other faster and more efficient optimization algorithms have emerged to train the generative model [46] [47].

Our framework can be decomposed in the following steps:

1. Creating the co-occurrence network from documents. We start with an empty network of words with no edges and for every document, we go through all unique pairs of words (i, j).

If there is no edge between (i, j) we add an edge with weight 1 and if an edge already exists we increment its weight by 1. Generating the network is therefore done in $\mathcal{O}(MN'^2)$ time, where N' is the number of unique words in a document. Depending on the definition of a document, N' is usually not very large and despite its quadratic term this step runs reasonably fast.

2. Finding the equivalent network for reflected random walk scheme. The reflection probability added after each step of the random walk turns the walker's first-order Markov chain into a second-order version. To keep computational benefits of first-order random walk models, we create a modified directed network on which the first-order random walk movements are an approximation to the reflected random walk on original undirected co-occurrence network. This modified network is constructed as follows:

   (a) First, for each node i we add a "reflection node" mirror i' that is to collect reflection movements outgoing from i.

   (b) We modify each outgoing edge weight w_{ij} to a new weight

   $$w'_{ij} = \left(\Big(1 - p_r(i \to j)\Big).w_{ij} \right)$$

   in the new network. If outgoing edge weights are normalized by their sum such that $\sum_j w_{ij} = 1$, this is the probability that the walker will go from i to j and does not reflect back.

   (c) The remaining portions of edge weights that correspond to reflecting back after the following step are aggregated and added as an edge from i to i': $w'_{ii'} = \sum_j p_r(i \to j)w_{ij}$.

   (d) We then either put a single edge back to i from i', or connect i' to the closest neighbors of i with equal weight (e.g. top 10 neighbors j of i whose PageRank is also very close to i: $_{PR(j)-PR(i)<\epsilon}$ for given ϵ).

On this new network, the walker will go from i to i' if it were about to reflect back in its next move. It then either goes back to i or one of its closest neighbors to mimic reflection.

It is worth noting the slight difference to the actual reflected random walk scheme, in that if the walker was about to visit a high-PageRank node j and reflect back, now this visit never occurs and the walker stays in the vicinity of i instead. This difference has little to no effect in our framework however, as we are extracting bottom-layer clusters after this modification and thus low-PageRank nodes are the part of the network we are focusing on.

To create this network, we first need to calculate PageRank of nodes. Although PageRank calculation via the classic power method [48], which finds the leading eigenvector of the modified network adjacency matrix, takes $|V|^2$ time, PageRank calculation is practically much faster for sparser networks and using more efficient algorithms [49]. Calculating PageRank is therefore one of the fast steps of our algorithm.

We are then adding $|V|$ reflection nodes with constant number of edges. In addition, each edge in the co-occurrence network needs constant number of operations, so creating the new network requires $\mathcal{O}(|V| + |E|)$ time and $\mathcal{O}(|V| + |E|)$ memory.

3. The next step is to find bottom-layer clusters. Infomap's optimization problem is to find a partitioning of the network that minimizes the entropy–based cost function described in section 6.1.1. It is impractical to check every partitioning in the network, so the implementation takes a greedy search approach along with simulated annealing to attack the optimization problem [35]. This implementation runs in $\mathcal{O}(|E|)$ time [50]. In our experiments, finding bottom-layer clusters is the computational bottleneck of the system and takes the longest time among these steps.

4. Once we have the n_c bottom-layer clusters, we need to generate the context network for each. We estimate contextual occurrence count for every node by assessing its $|\mathcal{C}|$ connections to cluster nodes, and update every edge to reflect contextual association. Since each edge is assessed no more than once for estimating contextual occurrence count of all outside words, creating a context network has time complexity $\mathcal{O}(|V| + |E|)$, so this step runs in $\mathcal{O}\big(n_c(|V| + |E|)\big)$ time.

5. The context-term matrix is formed by calculating the TF-ICF on terms for context profiles.

Calculating the ICF of every word needs $\mathcal{O}(|V|n_c)$ operations. Once these values are calculated, finding TF-ICF of every term in every context also takes $\mathcal{O}(n_c|V|)$ operations. In most practical examples $n_c \ll M = |D|$.

6. The next step is factorizing layer blocks $T \in \mathbb{R}^{n_c \times n_L}$ of the context-term matrix. The layered matrix factorization update rules described in section 6.2 take $\mathcal{O}(n_c n_L k_L)$ number of operations in one iteration. We slide a 3-bin layer window one bin at a time, so each word shows up in a fixed number of three layers. It is easy to derive the time complexity of running all layer factorization jobs is in the order of $\mathcal{O}(n_c n_v k_L)$, where $n_v = |V|$ is the number of words in vocabulary.

7. Similar to the layered factorization, the iterations in summary NMF require $\mathcal{O}(n_c n_v k)$ in each iteration.

CHAPTER 7

Conclusion

In this dissertation we proposed a new framework for understanding and extracting insights from large text corpora that aims to provide additional and more intuitive information to existing topic modeling methods. Most state-of-the-art approaches in topic modeling view topics as multinomial distributions over words. Our model, motivated by network models of semantic memory and other cognitive science models of memory's episodic nature, proposes a hierarchical structure of concepts that drive the process of writing documents.

Chapter 1 introduces the aforementioned models of semantic memory as random walks on an associative network of words. These studies which base their models on results of memory recollection experiments on human subjects, provide the intuition that forms the foundation of our work. The first challenge faced by directly applying the random walk memory model to our problem of modeling documents was that although writing documents is a process driven by systematic probing of memory, it is not entirely similar to the memory recollection tasks described and modeled in these studies, in the sense that document writing has more focused scope and context.

In Chapter 2 we analyzed the structure of an associative network of words built from word co-occurrence in a text corpus. We pointed out key differences that make document writing different from free memory recollection tasks and studied structural properties of the co-occurrence network to devise a suitable strategy for modeling document generation and finding a representation of the underlying knowledge structure. To address challenges proposed by contextual nature of documents, we also made use of works in psychology that model episodic memory and are thus able to describe contextual patterns in memory retrieval. The result was our new hierarchical view of concepts that form the knowledge base —with concepts seen as word clusters— and the notion that the very detailed and highly focused concepts define contexts. Writing a document in one of

these contexts imposes a bias on the associative network, or a bias on relative significance of words which we call the context profile, that affects memory retrieval.

Chapter 3 draws out the methodology to find the underlying hierarchical structure of concepts. We introduced a modified random walk scheme that traps the walker in context-specific clusters of words, in the presence of dominant *general words* in the associative network that disturb the walker's movement and make it randomly jump between contexts. We then used these context cores to build up the hierarchy by finding contextually significant concepts at different layers and the hierarchical connections among them. A key contribution of this method is the generalized shape of the hierarchy beyond simple trees of concepts or topics. We also showed how the collection of context profiles can be factorized to create summary topics similar to LDA topics.

These extracted summary topics for different datasets were then compared with LDA topics in chapter 4 to ensure conservation of topical information in our compressed representation of the text corpus in a context-term matrix. Layered concept clusters extracted from context co-occurrence patterns of higher layer words were also validated by examining their modularity in the document co-occurrence word network. We further elaborated on sample experimental results and examples of practical insights provided by the concept hierarchy in discussions of chapter 5. These test cases show how the topic composition of a text corpus and the subjective definition of a document result in diverse shapes of the hierarchy.

REFERENCES

[1] Zhou Z. Information Dynamics in Social Interactions: Hidden Structure Discovery and Empirical Case Studies. Los Angeles: Electrical Engineering Dept., UCLA; 2013.

[2] Joshua T. Abbott, Austerweil, Joseph L., and Thomas L. Griffiths, "Human memory search as a random walk in a semantic network." *Advances in neural information processing systems*, 2012.

[3] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research 3* (2003): 993-1022.

[4] Tangherlini, Timothy R., et al. ""Mommy Blogs" and the Vaccination Exemption Narrative: Results From A Machine-Learning Approach for Story Aggregation on Parenting Social Media Sites." JMIR public health and surveillance 2.2 (2016).

[5] Troyer, Angela K., Morris Moscovitch, and Gordon Winocur. "Clustering and switching as two components of verbal fluency: evidence from younger and older healthy adults." *neuropsychology* 11.1 (1997): 138.

[6] Hills, Thomas T., Michael N. Jones, and Peter M. Todd. "Optimal foraging in semantic memory." *Psychological review* 119.2 (2012): 431.

[7] Griffiths, Thomas L., Mark Steyvers, and Alana Firl. "Google and the mind predicting fluency with pagerank." *Psychological Science* 18.12 (2007): 1069-1076.

[8] Rhodes, Theo, and Michael T. Turvey. "Human memory retrieval as LÃ©vy foraging." *Physica A: Statistical Mechanics and its Applications* 385.1 (2007): 255-260.

[9] Harris, Zellig S. "Distributional structure." Word 10.2-3 (1954): 146-162.

[10] Nelson, Douglas L., Cathy L. McEvoy, and Thomas A. Schreiber. "The University of South Florida free association, rhyme, and word fragment norms." *Behavior Research Methods, Instruments, and Computers* 36.3 (2004): 402-407.

[11] Brin, Sergey, and Lawrence Page. "Reprint of: The anatomy of a large-scale hypertextual web search engine." Computer networks 56.18 (2012): 3825-3833.

[12] Deerwester, Scott, et al. "Indexing by latent semantic analysis." Journal of the American society for information science 41.6 (1990): 391.

[13] Nigam, Kamal, et al. "Text classification from labeled and unlabeled documents using EM." Machine learning 39.2 (2000): 103-134.

[14] Hofmann, Thomas. "Probabilistic latent semantic indexing." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

[15] Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing Order into Text." EMNLP. Vol. 4. 2004.

[16] Ravasz, Erzsébet, and Albert-László Barabási. "Hierarchical organization in complex networks." *Physical Review E* 67.2 (2003): 026112.

[17] Yee, E, et al. "Semantic Memory." The Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience, 4th ed., vol. 3, Wiley, 2017.

[18] Hintzman, Douglas L. "MINERVA 2: A simulation model of human memory." Behavior Research Methods, Instruments, & Computers 16.2 (1984): 96-101.

[19] Kwantes, Peter J. "Using context to build semantics." Psychonomic Bulletin & Review 12.4 (2005): 703-710.

[20] Ding, Chris, Tao Li, and Wei Peng. "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method." AAAI. Vol. 6. No. 42. 2006.

[21] Unser, Michael. "Splines: A perfect fit for signal and image processing." IEEE Signal processing magazine 16.6 (1999): 22-38.

[22] Griffiths, Thomas L., et al. "Hierarchical topic models and the nested chinese restaurant process." Advances in neural information processing systems. 2004.

[23] Tibshirani, Robert. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society. Series B (Methodological) (1996): 267-288.

[24] Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent." Journal of statistical software 33.1 (2010): 1.

[25] Ahluwalia, Ansuya, et al. "An automated multiscale map of conversations: Mothers and matters." Social Informatics (2012): 15-28.

[26] Bache, Kevin, and Moshe Lichman. "UCI machine learning repository." (2013): 2013.

[27] Zhou, Zicong, et al. "Information resonance on Twitter: watching Iran." Proceedings of the first workshop on social media analytics. ACM, 2010.

[28] Lewis, David D., et al. "Rcv1: A new benchmark collection for text categorization research." Journal of machine learning research 5.Apr (2004): 361-397.

[29] Eighteenth Century Collections Online dataset. URL `https://quod.lib.umich.edu/e/ecco/`

[30] Digital Humanities Resources for Project Building — Data Collections and Datasets. URL `http://dhresourcesforprojectbuilding.pbworks.com/w/page/69244469/Data%20Collections%20and%20Datasets`

[31] Chang, Jonathan, et al. "Reading tea leaves: How humans interpret topic models." Advances in neural information processing systems. 2009.

[32] Newman, David, et al. "Evaluating topic models for digital libraries." Proceedings of the 10th annual joint conference on Digital libraries. ACM, 2010.

[33] Yi, Xing, and James Allan. "Evaluating topic models for information retrieval." Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008.

[34] Pons, Pascal, and Matthieu Latapy. "Computing communities in large networks using random walks." Computer and Information Sciences-ISCIS 2005. *Springer Berlin Heidelberg*, 2005. 284-293.

[35] Rosvall, Martin, and Carl T. Bergstrom. "Maps of random walks on complex networks reveal community structure." *Proceedings of the National Academy of Sciences* 105.4 (2008): 1118-1123.

[36] C.E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, July, October, 1948

[37] Zhang, Sheng, et al. "Learning from incomplete ratings using non-negative matrix factorization." Proceedings of the 2006 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2006.

[38] Lee, Daniel D., and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." Advances in neural information processing systems. 2001.

[39] Brunet, Jean-Philippe, et al. "Metagenes and molecular pattern discovery using matrix factorization." Proceedings of the national academy of sciences 101.12 (2004): 4164-4169.

[40] Wang, Guoli, Andrew V. Kossenkov, and Michael F. Ochs. "LS-NMF: a modified non-negative matrix factorization algorithm utilizing uncertainty estimates." BMC bioinformatics 7.1 (2006): 175.

[41] Cuthill, Elizabeth, and James McKee. "Reducing the bandwidth of sparse symmetric matrices." Proceedings of the 1969 24th national conference. ACM, 1969.

[42] Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. Mining of massive datasets. Cambridge university press, pp 7-10, 2014.

[43] Newman, Mark EJ, and Michelle Girvan. "Finding and evaluating community structure in networks." Physical review E 69.2 (2004): 026113.

[44] Newman, Mark EJ. "Modularity and community structure in networks." Proceedings of the national academy of sciences 103.23 (2006): 8577-8582.

[45] Newman, Mark EJ. "Analysis of weighted networks." Physical review E 70.5 (2004): 056131.

[46] Teh, Yee W., David Newman, and Max Welling. "A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation." Advances in neural information processing systems. 2007.

[47] Porteous, Ian, et al. "Fast collapsed gibbs sampling for latent dirichlet allocation." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[48] Golub, Gene H., and Charles F. Van Loan. "matrix computations, 3rd." (1996).

[49] Lehoucq, Richard B., Danny C. Sorensen, and Chao Yang. ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods. Society for Industrial and Applied Mathematics, 1998.

[50] Mukherjee, Animesh, et al. "Dynamics On and Of Complex Networks, Volume 2." AMC 10 (2013): 12.

[51] Salmon, Daniel A., et al. "Health consequences of religious and philosophical exemptions from immunization laws: individual and societal risk of measles." Jama 282.1 (1999): 47-53.

[52] Etkind, Paul, et al. "Pertussis outbreaks in groups claiming religious exemptions to vaccinations." American Journal of Diseases of Children 146.2 (1992): 173-176.

[53] Williams, Sarah E. "What are the factors that contribute to parental vaccine-hesitancy and what can we do about it?." Human vaccines & immunotherapeutics 10.9 (2014): 2584-2596.

[54] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).