DREXEL UNIVERSITY

Toward Effective Knowledge Discovery in Social Media Streams

by

Anton Slutsky

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
College of Computing and Informatics

April 2015

"There were 5 Exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days."

Eric Schmidt

DREXEL UNIVERSITY

Abstract

Advisor: Xiaohua (Tony) Hu, Ph.D.
Co-Advisor: Yuan An, Ph.D.
College of Computing and Informatics

Doctor of Philosophy

by Anton Slutsky

The last few decades have seen an unprecedented growth in the amount of new data. New computing and communications resources, such as cloud data platforms and mobile devices have enabled individuals to contribute new ideas, share points of view and exchange newsworthy bits with each other at a previously unfathomable rate. While there are many ways a modern person can communicate digitally with others, social media outlets, such as Twitter or Facebook have been occupying much of the focus of inter-person social networking in recent years.

The millions of pieces of content published on social media sites have been both a blessing and a curse for those trying to make sense of the discourse. On one hand, the sheer amount of easily available, real time, contextually relevant content has been a cause of much excitement in academia and the industry. On the other hand, however, the amount of new diverse content that is being continuously published on social sites makes it difficult for researchers and industry participants to effectively grasp.

Therefore, the goal of this thesis is to discover a set of approaches and techniques that would help enable data miners to quickly develop intuitions regarding the happenings in the social media space. To that aim, I concentrate on effectively visualizing social media streams as hierarchical structures, as such structures have been shown to be useful in human sense making [1, 2].

Acknowledgements

While, perhaps, many students enter a PhD program as a step towards their future carriers, my motivations were different. At the time I started my studies at Drexel, I had had many years of software engineering under my belt and was well established as a seasoned professional with a clear industry carrier path. While my ultimate focus on the industry has never wavered, I made a decision to pursue a PhD degree late in my Master's studies five years ago because I believed that deep understanding of the scientific process and the scientific advancements in the area of Machine Learning and Data Mining were becoming overwhelmingly necessary in the modern world. Considering the recent explosion of the adoption of Big Data and, more recently, emergence of the Internet of Everything, it appears that my early intuition is proving to be the correct one.

While my focus towards this thesis has been reinforced over the years by the apparent applicability of my research work to the changes taking place in the industry, my progress was never been an easy one. From the early days of my studies to the present, I have constantly been discovering gaps and omissions in my understand of theoretical concepts and principles as they became relevant to my work. In order to fill these gaps, I relied heavily on the learning skills and techniques that have been taught to me early on in the Drexel's PhD program. Specifically, I would like to thank Drexel's iSchool PhD program for an excellent set of required courses that stimulated creative thinking and helped me learn how to navigate the boundless ocean of scientific publications. I would like to thank professors Susan Gasson, Katherine McCain and other professors for tough, but rewarding courses.

But, while course work is surely important, it has often been mentioned to my PhD cohort by senior professors that PhD is not about course work – it is a mentorship effort. Luckily, the mentorship support I have received at the iSchool has been an excellent one. As luck would have it, my first academic advisor turned to be Dr. Yuan An. Notwithstanding my earlier misgiving about various theoretical concepts, Dr. An has, on many occasions, patiently and skillfully worked with me to guide me in the right direction. Dr. An's comments and corrections to my publication drafts have always been insightful and meaningful. I would like to thank Dr. Yuan An for his help, support and friendship and I hope that he continues to lead future students with the same passion and interested as the showed towards me.

While the help I have received from Dr. An and others has been invaluable, I attribute much of the credit for moving me along toward this thesis to my academic advisor, Professor Xiaohua (Tony) Hu. Under Dr. Hu's guidance, I was able to discover new

and interesting concepts in machine learning and discover new tools and techniques that have helped me both academically and professionally. Over the years, Dr. Hu has always been available to answer questions and help with various drafts and work efforts. His expert knowledge of data mining and artificial intelligence has been invaluable to me in my work towards this thesis. I am especially grateful to Dr. Hu for his frequent suggestions of papers and lecture videos as these papers and videos have – on many occasions – guided me to new ideas and new ways of looking at the problems I tackle in this thesis.

Last, but certainly not least, I would like to thank my wife Rita for her support over the last five years. It so happened that my entrance into the PhD program had coincided with the birth of my son Tavi, which was later followed by the arrival of my daughter Kara. As many parents may attest, first few years of a child's life are probably the most challenging ones and, while I focused on classes and research work as well as my professional carrier, Rita – who is also a seasoned technologist with a graduate degree and a carrier of her own – took the brunt of the sleepless nights and feeding schedules. Her never wavering support and faith over the years has been invaluable in helping me get through numerous experiment failures and paper rejections. Without Rita's support it seems unlikely that this thesis could have been written.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivations

1.1 Introduction

Recent years have seen an explosion in popularity of the social web. Microblog sites such as Twitter and Facebook among many others attract millions of active contributors that use these sites as a communication vehicle to stay in touch with friends and family as well as a medium for publishing their ideas, exchanging thoughts and voicing concerns. Wide availability of mobile computing in the form of mobile phones and other portable devices that are ubiquitous today further propagates the social web phenomenon, as individuals are now able to contribute content almost continuously. Tweets and Facebook updates submitted during music concerts, political demonstrations and even in the course of natural disasters are commonplace and have become almost an expected norm in todays society.

As social media is now deeply engrained in the fabric of modern society, it provides an important source of data for data mining applications. Businesses and government agencies alike target the social web content to discover peoples desires, thoughts and motivations. Therefore, in this thesis, I propose methods to improve performance of text mining of social stream data. I concentrate on leveraging graph structures to improve the quality of knowledge discovery in social stream. Further, to ensure usefulness of the proposed methods in real world applications, the algorithms proposed here take into account the voluminous nature of social web output and operate in a manner that allows for scalable data mining of continuous streams.

One important goal for social stream data mining system is to present a view of social discourse to decision makers in such a way as to enable them to quickly grasp the overall gist of conversations taking place in the social forums such as Twitter, Facebook and

others. Such a system should be the ability to quickly respond to changing consumer interests and adjust product and service offerings to better serve current and potential customers. In todays fiercely competitive business environment, ability to gain timely insights into publics desires and needs would offer clear competitive advantage to industry organizations.

Therefore, in my research I work towards constructing a system that would monitor social media messages and produce meaningful views of the discourse. The goals of such a system would be threefold. First, in order to be useful the system would have to operate continuously for extended periods of time. That is, the underlying algorithm would have to be scalable in terms of space complexity to avoid overwhelming operational memory resources. Second, the system needed to produce output as close to real-time as possible. That is, the time complexity of the underlying algorithm had to be low enough to allow output to be generated nearly instantaneously upon arrival of new input from social streams. Lastly, the output of the system had to be of high quality. That is, output produced by the system would need to be predictive of future messages on the same subject with reasonable accuracy.

1.2 Research Questions

In this thesis, I consider social streams from two distinct perspectives. First, I view social media discourse as being related to a set of underlying topics. That is, save for noise and chatter messages, I think of microblog utterings as being associated with some general themes (or topics), with each theme represented by a probability distribution over possible words. Considering this view of social blogs, implementers of a data mining system tasked with monitoring social stream messages may wish to learn the nature of these topics, their numbers, their makeup and the nature of associated probability distribution. As learning topic models from a never-ending stream of text is a challenging undertaking, this thesis attempts to answer the following research question:

• *How to conduct topic discovery in social streams (microblogs) in a scalable way while improving quality of topic modeling?*

Another way to think of social media discourse is as being of a set of interrelated concepts, with each microblog message being somehow related to one or more of these underlying concepts. Such a view of social discourse appears natural as concept graphs (or ontologies) are often used to represent real world phenomena. Therefore, a data mining system capable of relating microblog utterings to existing concept ontologies or one

that is able to discover these ontologies automatically would surely be of great interest. With that, my second research question for this thesis is:

- *How can concept graphs be used to represent social discourse in microblogs.*

For a system aiming to understand concept relationships within a streaming body of data such as social web discourse, it is important to be able to detect the set of underlying concepts (topics) as well as discover relationships between these concepts in a scalable way. Therefore, both research questions are related, as scalability of topic mining in microblogs is an essential requirement for linking topics with concept graph entities in a realistic setting.

1.3 Thesis Structure

While my goal is to understand how to effectively organize large volumes of social media data as intuitively understandable hierarchies of interrelated concepts, the task appeared to be overwhelming early in my research efforts because of the sheer volume and velocity of social data.

Therefore, Chapters 2-3 outline my initial steps towards my research questions that focused on improving data clustering techniques for simpler and smaller data sets that nevertheless share many similarities with social media data. While these techniques have been useful in various practical settings and have been shown to help researchers grasp short and noisy data in terms of well-structured hierarchies, it was immediately obvious that these early efforts, while helpful and useful in many respects, could not be applied to more general data sets, such as social media streams.

In subsequent chapters, I depart from simple clustering approaches and focus on statistical topic modeling techniques. In Chapter 4, I outline an extension to supervised topic modeling, which allows short and noisy data (attributes similar to social media) to be grasped as topics in a predefined hierarchy. Then, in Chapter 5, I further refined the topic modeling approach to operate on streaming data (as opposed to static data corpora) in a way that allows past stream histories to have an effect on topic modeling in a scalable way. Then, in Chapters 6 and 7, I outline an approach for a semi-supervised topic modeling approach for social streams that allows users to visualize social streams as hierarchies of subjects focused around areas of users' particular interest.

Chapter 2

Automatic Approaches to Clustering Occupational Description Data

Our initial approach was to consider streaming text data in context of some available hierarchy. We were motivated by the presence of a multitude of manually defined tree-like taxonomies as well as the reset increase in interest from the industry towards taxonomy management tools and expression formats. While our ultimate focus was to try to understand social media sites such as Twitter and Facebook, at the time of writing, no universally accepted well-organized taxonomy of microblogs were available. We therefore considered some of the properties of social media streams and searched for other sources of data that exhibits similar properties and also corresponds to some existing, well-defined taxonomy.

Specifically, we considered that messages in social streams are generally quite short. For example, we found that messages in the publicly available and commonly sited Tweets2011 data set [3] contained, on average, 12.3 words per message (including stop-words). In addition, microblog postings are noisy and often contain misspellings and other modifications that obfuscate the canonical form of some words.

Therefore, we began our study by considering other, non-microblog data sources in terms of the aforementioned properties and search for similar corpora that would be accompanied by quality taxonomies.

2.1 Workplace Exposure to Beryllium Data

One such data source appeared to be the collection of occupation health measurement records collected by the Occupational Safety and Health Administration (OSHA). This regulatory compliance data included records containing short free text job descriptions and associated numerical exposure levels. Researchers in public health domain often needed to map job descriptions to Standard Occupational Classification (SOC) nomenclature for estimating occupational health risks. Previous manual process was time-consuming and did not advance so far to linkage to SOC.

While our ultimate goal was to organize the short and noisy free text job descriptions according to the SOC taxonomy, our first essential step was to discover an effective clustering approach for the texts. We were motivated by the idea that, once quality clusters were discovered, they could be instrumental in mapping job descriptions with appropriate SOC terms.

Our study indicated that the Tolerance Rough Set with Jaccard similarity was a better combination overall. The utility of the algorithm was further verified by applying logistic regression and validating that the predictive power of the automatically generated classifications, in terms of association of job with probability of exposure to beryllium above certain threshold, closely approached that of the manually assembled classification of the same 12,148 records.

2.1.1 Introduction

To protect the health of workers in the work place, U.S. Department of Health and Human Services stipulates that exposure to various harmful agents should be limited and kept within safe limits [4]. To ensure compliance, Occupational Safety and Health Administration (OSHA) monitors exposure by conducting surveys and collecting data on potential harmful agents found in the workplace [5], including metals such as beryllium, which is an increasingly common, though still rare, exposure in US workforce [6] . Each such sample is typically collected in the immediate vicinity of an employee (by personal or area sampling) whose job description is then associated with the sample [5]. Figure 2.1 contains a representative example of data recorded on "jobs".

A recent study conducted by Hamm et. al. [7] used the sample data stored in OSHA Integrated Management Information System (IMIS). The objective of the study was to develop several job-exposure matrices (JEMs) and to devise an approach that could be used to predict the proportion of workers exposed to detectable levels of beryllium [7]. In order to arrive at the aforementioned matrices, the researchers examined 12,148

38	IRONWORKER
39	IRONWORKER
40	IRONWORKER
41	QUALITY ASSURANCE
42	FOREMAN AND WELDER
43	SHAKEOUT
44	AUTOMATIC MOLDER
45	COREMAKER
46	MELTER

TABLE 2.1: Sample OSHA IMIS records

measurement records. Records were classified based on the manually recorded job description fields of each IMIS record and grouped according to the type of occupation. For example, records with job description field containing strings green sand mold, automatic molder, and auto molder were classified as molder.

While the analysis of the resulting JEMs proved useful and highlighted various observable trends in beryllium exposure in the workplace, the classifying process itself was extremely tedious and exuberantly time consuming requiring several months of manual effort to complete. As exemplified in [7] as well as other studies that utilized the OSHA IMIS data [8, 9], the need to manually process thousands of free text entries may be an obstacle for future research attempts to study the data. In this paper, we attempt to reduce the tedious and non-reproducible task by investigating automated alternatives to clustering the job description data. We study the problem of assisting researchers in public health domain, occupational health in particular, to quickly create data that can be used to understand determinants of occupational (airborne) exposure.

We investigate different clustering methods in combination with several similarity measures and then test how automatically produced clusters compare manually assembled ones in predicting exposure to beryllium in OSHA IMIS data previously accessed by Hamm & Burstyn [7]. The IMIS job description records we studied in this research are quite short containing on average 2.217 terms per record. All 2,858 unique job descriptions were used in the experiment; 1,408 of these records contained top 10 ranking terms. While this is hardly surprising considering the well-known term distribution empirical laws [10], with each term accounting for approximately half of all terms in a record, the highly repetitive nature of frequent terms appears to be a useful property of the data. The remainder of this chapter is organized as follows. In Section 2.1.2, we describe the research methods, including algorithms, similarity measures, and data processing. In Section 2.1.3, we report the experimental results before and after improvements. In Section 2.1.4, we discuss related work. In Section 2.1.5, we present our conclusions.

2.1.2 Clustering Methodology

In order to identify an effective unsupervised classification approach for the type of data found in OSHA IMIS records, we experimented with a number of well-known as well as recently introduced clustering algorithms. We combined these algorithms with various similarity measures and compared the results of each combination.

2.1.2.1 Clustering Algorithms

In this study, we consider the following clustering algorithms: *Tolerance Rough Set algorithm*, K-Mean Clustering, ROCK Clustering Algorithm, and CHAMELEON Clustering Algorithm.

Tolerance Rough Set Algorithm: Clustering based on tolerance rough set model uses notation postulated by the rough set theory an extension of the general set theory [6]. We adopt the algorithm developed in [11].

K-Means Clustering: The commonly used partitioning algorithm is K-Means clustering [12] which initially selects k points inside the hyper-volume containing the data set. It then assigns each data pattern to the nearest cluster center, and updates cluster centroids using current cluster membership until convergence criteria have been met.

ROCK Clustering Algorithm: ROCK algorithm [13] is an example of bottom-up (or agglomerative) clustering approach. As all other agglomerative algorithm, ROCK starts off by partitioning data into large number of clusters and proceeds to merge these clusters until a desired number of partitions is reached. ROCK differs from other hierarchical algorithms in that it provides an innovative heuristic for identifying best merge candidates at each level of agglomeration.

CHAMELEON Clustering Algorithm: CHAMELEON clustering approach improves upon ROCK by measuring cluster similarity using a dynamic model. Its key feature is that it judges cluster similarity by taking into account both the inter-connectivity as well as the closeness of the clusters. [14]

2.1.2.2 Similarity Measures

Each of the clustering algorithms described above was combined with a number of commonly used metric and non-metric similarity measures.

Metric Similarity Measures (Euclidean Distance): To compute the Euclidean distance, we consider each job description as a document. For example, let $A =' foo'$

and $B =' bar'$ be two job description documents. The vector space for these documents would then be a set $V_{AB} = \{`foo', 'bar'\}$. We defined the document vectors for A and B as $\vec{V}_A =< 1, 0 >$ and $\vec{V}_B =< 0, 1 >$ respectively using a binary weight function. Having converted string documents into geometric coordinates, we can compute the distance between the vectors.

N-gram Similarity Measure: N-gram similarity, as described by [11], is a process of identifying the length of the longest common sub-sequence (LCS) between two strings. The sub-sequence is constructed by finding matching n-grams in the two strings with respect to the order of occurrence.

Jaccard Coefficient: Jaccard coefficient relies on a set-theoretic view of candidate strings. It first converts strings into sets of terms and then measures the overlap between the sets defining the coefficient J as [13]:

$$J_{A,B} = \frac{A \cap B}{A \cup B} \tag{2.1}$$

2.1.2.3 Pruning and Normalization

IMIS OSHA exposure record data exhibits several important properties such as brevity and lack of excessive data noise. IMIS job descriptions were tokenized and terms containing non-letter characters where removed. For example, string 1st shift operator was normalized to shift operator by removing the term 1st which contained a number. Once the extraneous strings were removed, remaining terms were stemmed using a Porter stemming algorithm [15].

2.1.3 Experimental Study and Methodology

Our goal was to cluster the raw job descriptions from the labeled data file used by Hamm & Burstyn [7]. Our evaluation was performed using the standard criteria: recall, precision, and F-measure. We took the labeled job description records as the resource for the gold standard. To create such a "gold standard", data items were grouped according to their corresponding labels. Resulting partitions were then regarded as pristine and used to calculate evaluation measurements.

It is important to note that the "gold standard" partitions were only assumed to be correct. Considering the tedious nature of the manual classification effort, some degree

of human error is likely. Thus, caution was exercised when drawing conclusions based on the accuracy measurements inferred from the labeled data.

Precision and recall were computed as follows. Let $C = \{c_1, ..., c_n\}$ be a set of clusters, $T = \{t_1, ..., t_m\}$ be a set of topics (groups of data items in the "golden standard" set) and $D = \{d_1, ..., d_l\}$ be a set of terms. Then, let D_c^i be a set of data items in cluster $c_i \in C$, D_t^j be a set of data items in topic $t_j \in T$ and $D_{c,t}^{i,j} = D_c^i \cap D_t^j$. Precision is defined as in [16]:

$$Precision(c_i, t_j) = \frac{|D_{c,t}^{i,j}|}{|D_c^i|} \tag{2.2}$$

and recall as:

$$Recall(c_i, t_j) = \frac{|D_{c,t}^{i,j}|}{|D_j^t|} \tag{2.3}$$

Having thus defined precision and recall in cluster context, corresponding F measure can be expressed as:

$$FMeasure(c_i, t_j) = \frac{2 * Precision(c_i, t_j)}{Recall(c_i, t_j)} \tag{2.4}$$

To evaluate performance across all clusters, we used an overall evaluation F-Measure defined in [16] as:

$$F = \sum_{c_i \in C} \frac{|D_c^i|}{|D|} max_{t_j \in T} FMeasure(c_i, t_j) \tag{2.5}$$

2.1.3.1 Initial Results

Raw job descriptions from the labeled data file were clustered using all combinations of similarity measures and clustering algorithms discussed in the previous section. Each similarity/cluster pair performance was then evaluated using the F-measure evaluation criteria. In order to provide the best possible comparison, each clustering algorithm described in section 2.1.2 was supplied with threshold values empirically determined to maximize the F-measure value in the context of each of the similarity functions.

Table 2.2 shows results of our first-round experiments. The best overall result, with the F-measure value of 0.42, was produced by combining the Tolerance Rough Set (TRS)

algorithm with the Jaccard Coefficient similarity measure. While the number appears to be quite low, evaluation of individual measures for larger classifications is encouraging. In particular, the largest cluster 'welder' was produced with the F-measure value of 0.8, with 0.8 and 0.81 precision/recall values, respectively. Precision and recall measurements for other clusters, such as 'polisher', exhibited similarly high degree of accuracy. High recall numbers (suggested in [16] to be indicative of quality clustering classification) were produced for 'operator' (0.95) and 'driver' (0.93).

	Jaccard Coefficient	**N-Gram (Bigram)**	**Metric (Euclidean)**
K-Means	N/A	N/A	0.34
ROCK	0.03	0.08	0.23
Chameleon	0.22	0.19	0.24
Tolerance Rough Set	0.42	0.37	0.15

TABLE 2.2: F scores of clustering algorithm and similarity measure combinations

2.1.3.2 Observations and Improvements

Contextual Pruning: Our analysis suggested that the TRS algorithm needed to take into account the variable importance of individual terms. Some terms (such as 'operator', 'worker' and 'helper') appeared to be less valuable in some cases, but equally valuable in others. For example, job descriptions such as 'mill operator' and 'welder helper' were manually classified as 'mill' and 'welder' respectively. On the other hand, strings 'vac operator' and 'saw operator' were labeled as 'operator'.

Observation that terms 'mill' and 'welder' occurred more often (100 and 4049 records respectively) in the set of records compared to 'vac' and 'saw' (3 and 13 records respectively) suggested that a weighting scheme could be useful in determining the importance of individual terms. This is in accord with intuition employed in manual coding where rare terms lead generally to small clusters of observation that contain insufficient information to make any reliable inferences about association with occupational exposure (here: beryllium concentration in workplace atmosphere).

One common approach to term weighting is through the concepts of term frequency and inverse document frequency. Term frequency tf is defined as the number of times a term occurs in a document divided by the total number of terms, while inverse document frequency idf is the ratio of the total number of documents in the corpus to the number of documents containing a given term. Using the aforementioned concepts, terms are often weighted by combining term frequency and inverse document frequency in a formula $w_t = tf_t \times idf_t$, where w_t is the weight of the term t and tf_t and idf_t are term frequency and

inverse document frequency of t respectively. Such weighting scheme rewards important terms while at the same time scales down the score of ubiquitous ones.

Our first attempt to improve the clustering results was to use type weighting. We converted records to weighted term vectors with weights. Cosine coefficient [17] was then used to assess similarity between the weighted vectors. Unfortunately, this approach did not produce any improvement. Further evaluation suggested that this result was not without basis. Examining document frequency counts for terms 'welder' and 'operator' pointed out that while 'operator' was the most frequent term (lowest idf), 'welder' was an incredibly close runner up (with only a decimal digit difference in their idf valued). With term frequencies weighing in quite heavily in short documents, $tf \times idf$ term boosting did not seem likely to produce the desired effect.

We developed a new approach that allows for the necessary flexibility. Since it seemed unlikely that the necessary context could be inferred from the data directly, a controlled set S_{prune} was compiled which contained terms deemed likely to require contextual processing. This list was then used to conduct contextual pruning of the representative R_k relation.

The pruning was conducted by constructing a set R_{prune} such that

$$R_prune = \begin{cases} 1 & \text{if } S_{prune}, \exists_{t \in R_k}(df(t) > \gamma) \\ \emptyset & \text{otherwise} \end{cases} \tag{2.6}$$

where t is a term, df is the document frequency function of term t and γ is a threshold number. Set R_{prune} was then subtracted from R_k and the difference set was used as the representative relation for a cluster.

2.1.3.3 Second-Round Results

We conducted our second-round experiments using the pruning algorithm above. The algorithm produced a noticeable improvement in the overall quality. The overall F-measure value increased by 28.6% from 0.42 to 0.54. Individual cluster measures were even more encouraging. The larger 'welder' cluster accuracy increased from F-measure value of 0.82 to 0.91, a 10% improvement. The accuracy of the cluster 'bencher' more than doubled from 0.42 to 0.85. Overall, with the addition of the contextual pruning modification, most of the clusters deemed valuable in the original Hamm & Burstyn study improved considerably. Figure 2.1 shows the distribution of improvements for clusters deemed useful in [7].

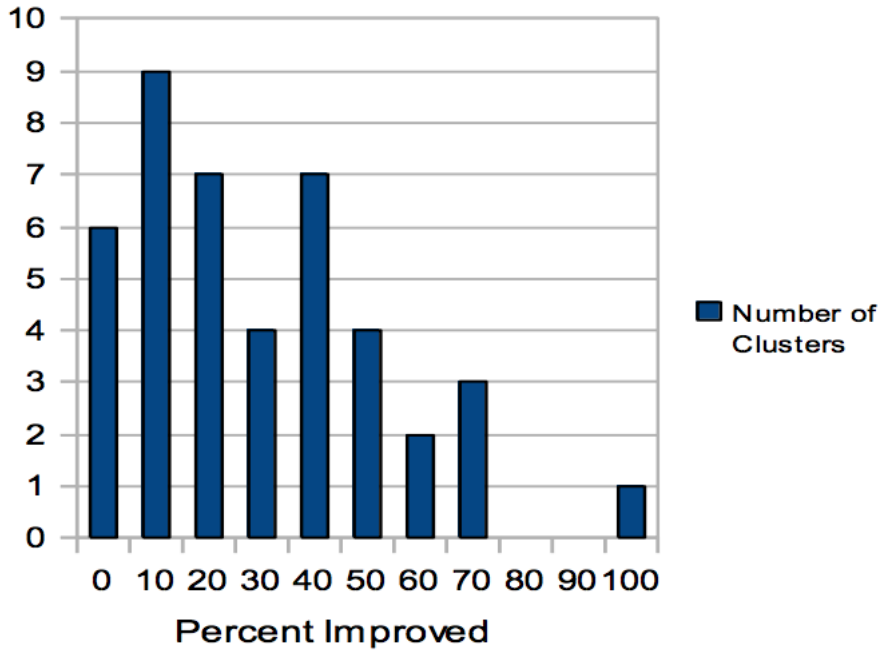


FIGURE 2.1: Improvement frequencies for useful clusters

In addition to individual precision/recall measurements, we also analyzed raw frequency counts of individual observations (and measurements of exposure to beryllium) for corresponding classifications compiled for the Hamm & Burstyn [7] study. These counts accentuated the observation that, while some precision/recall numbers are low, measurements for the larger clusters that make up a large portion of the data performed well. For example, manually assembled classification 'welder' contained 3921 records comprising 32% of the 12,148 records.

Hypothesizing that larger clusters would be more accurate, linear regression was used to test this conjecture. Figure 2.2 relates F-measure values to frequencies of corresponding clusters as counted in the original study.

Slopes of the regression lines in Figure 2.2 indicate that there exists a positive correlation between sizes of clusters and automatic classification accuracy. This association is important to note because, intuitively, there are more data associated with frequently occurring jobs in any sampling analysis by virtue of those jobs being common. High precision/recall numbers for clusters that encompass large portions of the data suggest that the quality of the automatically generated classification clusters approached that of classifications created manually.

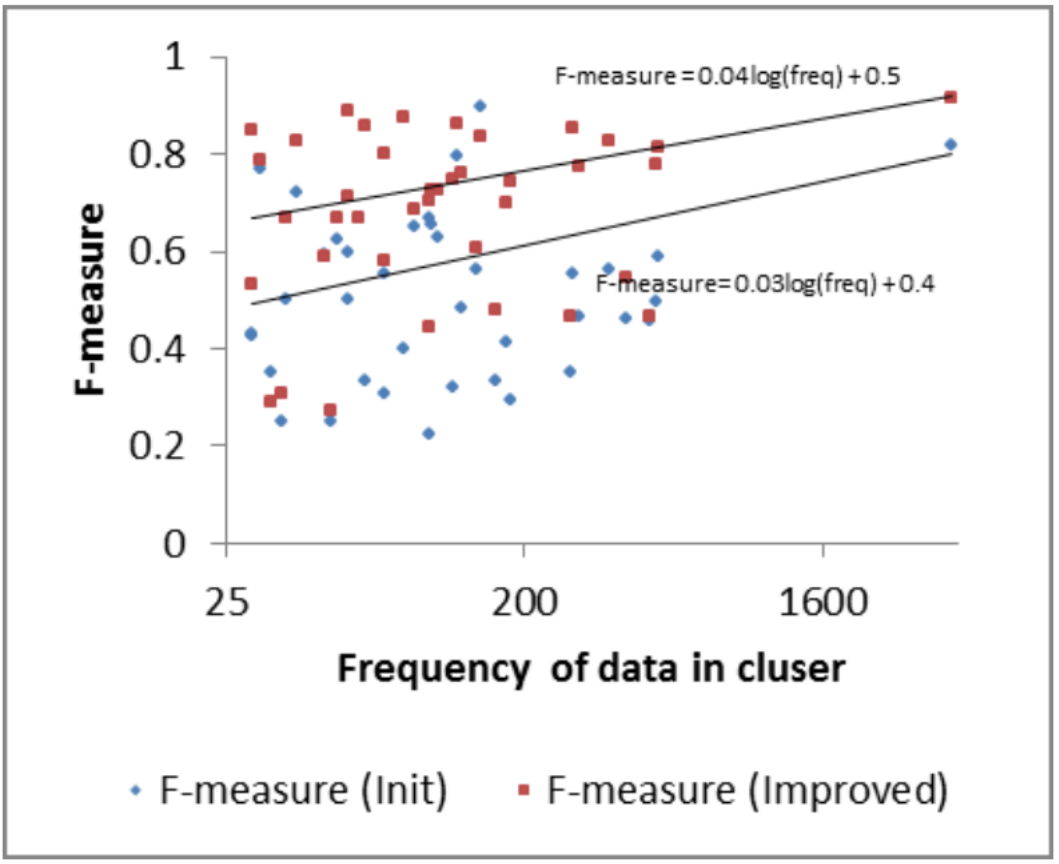


FIGURE 2.2: Frequency (log scale) vs. F-measure of useful clusters; fitted least square regression equations shown

2.1.3.4 Utility Assessment

In order to judge the usefulness of the improved algorithm, the full set of the 12,148 measurement-job pairs of records was automatically clustered. The generated clusters were then related to recorded exposure to beryllium in the same way as the manually assembled classifications from the original study [7]. Logistic regression was used for evaluation of the association between jobs (manually coded or produced automatically) and probability of exposure to beryllium above thresholds deemed to be important for risk assessment [7]. Here, as in [7], clusters were judged useful if they contained more than 30 records.

Using the logistic regression model (PROC LOGISTC in SAS v9, SAS Institute, Cary, NC) predictive accuracy of the automatically generated clusters was then compared to that of the manually compiled classifications. The comparison was conducted using several measures that estimate how well the regression model predicts the data. These measures were R^2, $Max-rescaled R^2$ and the Akaike Information Criterion (AIC). R^2 is a ratio of variation explained by the model to the observed variation. $Max-rescaled R^2$ refines R^2 by allowing for the value of 1 to be reached. The AIC measure is commonly

used to estimate the fit of a model. Predictive ability of manual and automatically compiled groupings are presented in Table 2.3

Clustering method	Number of clusters*	Definitions of predicted exposure modeled in logistic regression					
		P[$Be > 0.1$]			P[$Be \geq 0.05$]		
		R^2	Max-rescaled R^2	AIC	R^2	Max-rescaled R^2	AIC
Manual	40	0.08	0.19	6165	0.07	0.20	4374
Automatic	51	0.08	0.18	6245	0.07	0.19	4441

TABLE 2.3: Predictive power of selected groupings

*In both cases, only groups containing more than 30 were considered in the evaluation.

Comparison of the automatic and manual classifications shows that the classifications generated using the optimized clustering algorithm closely matched the predictive power of the manually compiled classifications. For example max-rescale R2 for manual and automated coding indicated that the two clustering methods both explain roughly 20% of variance in probability exposure for either threshold. While the manual coding resulted in slightly better predictions, the difference is not sufficient to justify the laborious effort.

2.1.4 Related Works

In a recent work, Obadi, et al. [18] used Tolerance Rough Set clustering algorithm to cluster records published by the Digital Bibliography & Library Project (DBLP). Obadi, et al. [18] conducted a comparative study evaluating the Tolerance Rough Set-based approach against other frequently used algorithms. Kumar, et al. used the Tolerance Rough Set method to successfully classify web usage data available through the UCI Machine Learning Archive [19]. Shan et al. use Rough Set theory to introduce a knowledge discovery approach aimed at identifying hidden patterns and transforming information into a simplified, easily understood form [20]. While this work does not utilize a clustering approach relying instead on a rule-based algorithm, the utility of the Rough Set-based method is exemplified by considering a set of vehicle records where makes and models are contained in the same field. Chen, et al. [21] used the Rough Set model to introduce a clustering algorithm aimed at grouping categorical data. Since the values of attributes of categorical data are restricted to sets of categories, categorical records are noise-free by design, but may contain missing values. Chen, et al. [21] compared their Rough Set based clustering algorithm to other popular clustering approaches and found the Rough Set approach suitable for clustering categorical data. While OSHA IMIS records are not categorical, the nature of categorical attributes suggests that high document frequency counts are to be expected in a categorical data set feature similar to that of the OSHA IMIS data.

2.1.5 Conclusions

We studied automatic approaches for clustering job description data provided by the OSHA IMIS database. Our experimental results suggested that the Tolerance Rough Set approach is a good candidate for the clustering task. We conducted experiments and compared results produced by the algorithm to those generated by human classifiers. Our results showed that the algorithm can be augmented with a list of terms for contextual pruning.

Our study highlighted a promising direction towards development of an automatic approach that would helpful to eliminate or significantly reduce manual effort required to map the short and noisy Occupational Health and Safety measurement descriptions into the well-structured SOC taxonomy. Unfortunately, the task of organizing the data into the SOC hierarchy remained a manual effort. In the following chapter, I will outline an approach towards automating this task further.

Chapter 3

Automatic Approaches to Classifying Occupational Health Records for the SOC Taxonomy

In this chapter, we continue to drive towards the development of an effective classification mechanism for short and noisy unstructured data such as the occupational health data. While occupational health is a narrow domain, because the observations about occupational health are short, unstructured and noisy, the approach proposed in this chapter serves as a stepping stone towards discovering a effective sense-making mechanism for other domains, such as social media, which is the ultimate goal of this thesis.

To that end, I investigated automatic approaches for classification of unstructured text data that describes occupations related to numerical observations for public health research. Data that motivated this particular work was collected during observational studies such as the the Womens Health Initiative includes records containing short, free text job descriptions attributed with numerical values. Researchers in public health domain need to map job descriptions to the Standard Occupational Classification (SOC) nomenclature in order to estimate risks to health due to occupational exposures (e.g. physical demand). Previously, the mapping was accomplished with a time-consuming and tedious manual process. We investigated alternative automatic approaches for classifying free text job descriptions. The classification results are an essential step towards automating the SOC matching process. Our study indicated that choosing classification with the lowest joint information content resulted in improvement of 155% most detailed hierarchy level and 125% next most detailed level in terms of precision measurements of the target hierarchy compared with the baseline state-of-the-art classifier. The utility of

the algorithm was further verified by validating that the average percent error in associated physical demand measure for the proposed approach was reduced by 76% versus the baseline classifier

3.1 Introduction

United States Bureau of Labor Statistics classifies workers according to the Standard Occupational Classification (SOC) system [22]. This nomenclature is used by various agencies during collection, analysis and dissemination of data. The SOC system is a hierarchical representation of occupations, with higher level occupational groups containing detailed nested subgroups. The goal of this hierarchy is to ensure that every occupation can be classified using a clearly defined nomenclature and to ensure that similar occupations are cluster [22].

The data organized according to SOC and made available by government and private agencies has been a valuable resource for academic investigations in various fields, including public health research. For example, the Womens Health Initiative (WHI) observational study [23, 24] is a multi-ethnic cohort of 93,676 postmenopausal women, 50 to 79 years of age, enrolled from 1993 to 1998 at 40 geographically diverse clinical centers throughout the United States. Ultimately, the WHIs goal was to identify and prevent the major causes of death, disability and frailty in older women of diverse socioeconomic and racial backgrounds. At baseline, participants provided detailed information about occupational history, other socio-demographic characteristics, medical history, and health behaviors through a self-administered questionnaire. Included in this study were women who have valid data on up to three main occupations (approximately 82,000 women). Participants were asked about the three paid jobs (full-time or part-time) held the longest length of time since they were 18 years old. For each job, the job title and industry where job was performed were elicited as free text as well as the age she started work and total duration of employment. Summarizing occupational histories in terms of SOCs would enable linkage of WHI data to other databases with information of characteristics of work, thereby enabling investigations of associations of womens occupational history and health.

In another application , in order to ensure compliance with workplace safety laws, US Occupational Safety and Health Administration (OSHA) conducts surveys and collects air samples to monitor exposure to potentially harmful contaminants in the workplace [25]. The air quality data is published through OSHAs Integrated Management Information System (IMIS). Numerous public health researchers [7, 9] have used this data in studies of harmful exposure patterns within various occupations. Their work identified

valuable insights into potential dangers of occupations. These insights may be instrumental in avoiding or preventing health related issues that arise from being exposed to various disease causing agents. However, more complete use of that data has been hindered by the fact that description of jobs is recorded as free text in IMIS.

Unfortunately, the valuable data on description of jobs is often stored in a form that makes it difficult for researchers to process. Records are associated with a manually keyed job description strings and are not coded with any standardized nomenclature in the two motivating examples presented above. In order for public health researchers to be able to produce meaningful results, potentially tens of thousands of manually entered job description strings must be painstakingly matched with the SOC classification codes. As reported in [7], this labor intensive manual effort may be a hindrance to future public health research in this area.

Considering the lengthy, tedious and resource intensive nature of matching unstructured job descriptions to standard SOC codes, using an automated classification approach is desirable. Classification of unstructured data is traditionally accomplished by associating documents with a relatively small set of well-defined labels [26]. This target label set is generally flat and often overgeneralizes the true conceptual model of the data. Such conceptual models are often represented as taxonomies, which are hierarchically grouped sets of concepts. One example of such taxonomy is the SOC system. Many novel techniques have been proposed in recent years that extend traditional unstructured data classification methods and take advantage of structures intrinsic to taxonomies [27].

Incorporating hierarchical taxonomies into data classification approaches has been shown to be necessary in many applications such as query classification, contextual advertising and web search improvements [27, 28]. Their usefulness notwithstanding, creating a successful taxonomy-based classifier model is not a trivial task. The difficulty arises from the fact that many real-world taxonomies are quite large and often do not expose large number of training documents associated with each hierarchical label [27]. Further, while many large taxonomies, such as those published by Yahoo! or ODM , expose millions of documents per category that could be used for classifier training, distributions of these documents are often skewed covering some hierarchical categories well and leaving others with few training example [27].

Many recently developed classification algorithms attempt to overcome the problems of limited training sets and prohibitively large hierarchies and strive to reap the benefits of applying taxonomy to unstructured data. Initial approaches to this type of classification made use of traditional facilities and viewed the target hierarchical taxonomy structure as a flat label set where each label was a hierarchy category. Unfortunately, this initial approach proved unsatisfactory [29]. Classifiers using this approach took excessively

long time to be trained and produced progressively worse classification results as sizes of taxonomies increased [27].

Current state-of-the-art taxonomy classification approaches improve on the nave algorithm by classifying data in two stages search state and classification stage. During the search stage, an inverted index is built from the taxonomy. The document being classified is then used as a query to search the inverted index. The search procedure produces a narrow set of relevant candidate categories which are then used in the classification stage during which classification model is trained for each new test document using the narrow (pruned) taxonomy sub-tree. The model is then used in conjunction with the test document and the document is labeled with the most probable class candidate category [27, 28].

While approaches based on the above description have been shown to produce significantly better results than previous implementations, one significant drawback of the current state-of-the-art machinery is the restriction that each document being classified must belong to a candidate category produced by the search stage. That is, current algorithms provide no way to take the hierarchy itself into account and use the branching structure of target taxonomies as a way to distinguish between lower-level candidate headings. This may be a problem in some contexts. In particular, while the search procedure often produces relevant category matches for documents, in some applications document content may not be sufficient to definitively lay claim to a low level hierarchy item.

The main drawback of the state-of-the-art algorithms is treating the hierarchical path of candidate categories as support objects rather than first-class candidates available for classification. That is, the set of possible target categories is limited to those located during the search stage. That may be problematic in some contexts where it may be necessary to generalize the final document category to higher levels of the hierarchy if the choice of hierarchy becomes arbitrary below a certain hierarchy level. To illustrate the problem, consider a vignette hierarchy of the SOC classification system in Figure 3.1 below.

If the algorithm were to classify a job description with a single term secretary, it would have to make a choice between Medical Secretaries, Legal Secretaries and Executive Secretaries and Executive Administrative Assistants. Since the algorithm must classify the input document into a single candidate category, the final choice would be an arbitrary one as any choice is equally plausible given the information in the document.

In this chapter, we discuss our attempts to improve the existing approaches and propose a novel algorithm for hierarchical classification. As opposed to other existing approaches,

43-0000 OFFICE AND ADMINISTRATIVE SUPPORT OCCUPATIONS
43-6000 Secretaries and Administrative Assistants
43-6011 Executive Secretaries and Executive Administrative Assistants
"Executive Secretary"
"Executive Assistant"
43-6012 Legal Secretaries
"Law Secretary"
"Legal Administrative Assistant"
43-6013 Medical Secretaries
"Hospital Secretary"

FIGURE 3.1: Sample Hierarchy quoted text (e.g..: Executive Secretary) represents training documents associated with corresponding categories.

the proposed algorithm improves upon existing hierarchical classification machinery by allowing the matching process to generalize classification decisions to different levels of target hierarchies. We test the utility of the proposed algorithm by applying it to a research problem from epidemiology and compare results with those produced by the baseline algorithm discussed in [27]. Our results show that average percent error in utility scores (discussed below) was 15.95% for the baseline algorithm and 3.8% for the novel algorithm presented here. Further, the proposed method improved the baseline by 155% detailed occupation and 125% broad occupation precision measurements for each level of the target hierarchy (Table 3.1).

The rest of the chapter is organized as follows. Section 3.2 discusses related work. Section 3.3 presents the proposed method. Section 3.4 describes results of applying the approach to classifying free-text job descriptions in WHI observational study against the SOC nomenclature. Section 3.5 discusses the advancements made to the previously developed approaches and limitations of the proposed method. Finally, Section 3.6 discusses future directions and concludes the chapter.

Level	**Precision Baseline**	**Precision Extended**	**% Improved**
Detailed Occupation	0.2	0.51	155%
Broad Occupation	0.32	0.72	125%
Minor Group	0.84	0.85	1%

TABLE 3.1: Precision values for SOC gradations using existing (baseline) method and newly proposed approach (extended).

3.2 Related Works

Generally, hierarchical data classification tasks are accomplished in several ways. The so-called *big-bang* approach treats the entire hierarchy as a set of target labels and trains the classifier on the entire document collection at once [30]. While many classification algorithms have been used to construct big-bang style classifiers [31, 32], these classifiers

were shown to take excessively long time to train and building such classifiers for very large hierarchies with hundreds of thousands of categories was intractable [33].

Top-down classification algorithm approaches the classification problem by training a separate classifier for each level of the hierarchy. While this approach solves the scalability problem of big-bang classifiers, it suffers from issues related to error propagation as misclassifying document high in the hierarchy chain implies no chance of getting the classification right at lower levels.

In a recent study, significant performance gains were achieved by a narrow-down process that conducts classification in two stages. First, for a given document to be classified, the hierarchy training set is searched using a similarity measure, such as cosine similarity. Then, the top k hierarchies closest to the input document vis--vis the similarity measure are said to be candidate hierarchies. The hierarchy tree is then pruned to contain only the candidate categories and any additional categories that are deemed necessary by the implementing algorithm. Hence the process narrows down the number of categories that would be used for training.

Identifying the set of categories that would assist in the final classification determination is the focus of algorithms that implement the narrow-down paradigm. Since one of the major problems of hierarchical classification is lack of training examples for all categories in the hierarchy, some algorithms expand the training set to include all documents belonging to candidate categories as well documents belonging to parents of candidate categories (ancestor-assisted). Others enhance the training pool by including both the ancestor-assisted documents and the immediate neighborhood of candidate categories. These latter approaches are referred to as neighbor-assisted approaches.

A recent improvement on the narrow-down approach was implemented by including top-level category information. The top-level (global) information is represented in a form of a category-level language model and the model is combined with the candidate-level (local) language model information via the use of a fixed mixture model parameterized by mixture weights.

3.3 Proposed Method

In this work, we attempt to extend the top-level (global) and candidate-level (local) mixture model hierarchy classification approach in [33] as it was shown to produce better results when compared to other common approaches. Similarly to [31, 32] and [33], the proposed approach starts by going through a search stage. The search stage

is accomplished by searching an inverted index of the training hierarchy. As in [31, 32] and [33], we used the Lucene search engine[1] as it is fast and freely available.

The inverted index was constructed by indexing leaf categories as individual documents and processing non-leaf categories by representing them via synthetic documents comprised of all documents associated with the descendant categories.

The index was subsequently searched for each input document in order to narrow down the hierarchy tree. The resulting set of candidate categories was then classified using a Nave Bayes Classifier (NBC). Let $C = c_1, , c_n$ be a set of n nodes in a taxonomy. The classifier estimated the posterior probability of a candidate category by

$$\begin{aligned} P(c_i|d) &= \frac{P(d|c_i)P(c_i)}{P(d)} \propto P(c_i) * \prod_{j=1}^{N} P(t_j|c_j)^{v_j} \\ P(t_j|c_i) &= (1-\gamma_j)P(t_j|c_i^{global}) + \gamma_j P(t_j|c_i^{local}) \\ P(c_i) &= (1-\gamma_j)P(c_i^{global}) + \gamma_j P(c_i^{local}) \end{aligned} \tag{3.1}$$

where $c_i \in C$ is the ith candidate category with $1 \leq i \leq n$ being an index of the category in C, d is an input document being classified, t_j is the jth term in document d, N is the size of vocabulary, and v_j is the number of time the jth term occurred in document d. Further, c_i^{global} is the top level category of c_i and c_i^{local} is the same as c_i but rephrased for explicit explanation, i.e. $c_i^{global} = A$ and $c_i^{local} = A/B/C$ for $c_i = A/B/C$ and $0 \leq {}_j \leq 1$ is the mixture weight. The probability of $P(c_i^{global})$ and $P(c_i^{local})$ are estimated as

$$\begin{aligned} P(c_i^{global}) &= \frac{|D_i^{global}|}{|D|} \\ P(c_i^{local}) &= \frac{|D_i^{local}|}{|D|} \end{aligned} \tag{3.2}$$

where D is the entire document collection and D_i^{global} is a sub-collection of documents in c_i^{global}, D_i^{local} is a sub-collection of documents in c_i^{local}. $P(t_j|c_i^{global})$ is approximated as a mixture

$$P(t_j|c_i^{global}) = (1-\alpha)P_{global}(t_j|c_i^{global}) + \alpha P_{global}(t_j) \tag{3.3}$$

[1] http://lucene.apache.org

where $0 \leq \alpha \leq 1$ is the mixture weight and

$$P_{global}(t_j|c_i^{global}) = \frac{\sum_{d_k \in c_i^{global}} tf_{jk}}{\sum_{t_u \in V^{global}} \sum_{d_k \in c_i^{global}} tf_{uk}}$$
$$P_{global}(t_j) = \frac{\sum_{d_k \in D} tf_{jk}}{\sum_{t_u \in V^{global}} \sum_{d_k \in D} tf_{uk}} \tag{3.4}$$

where tf_{ji} is the term frequency of term t_j in top-level category c_i^{global} and V^{global} set of terms selected by the chi-square feature selection method over the entire document collection and

$$P(t_j|c_i^{local}) = \frac{\sum_{d_k \in c_i^{local}} tf_{jk}}{\sum_{t_u \in V^{local}} \sum_{d_k \in c_i^{local}} tf_{uk}} \tag{3.5}$$

where tf_{ji} is the term frequency of term t_j in top-level category c_i^{local} and V^{local} set of terms among D.

Here, document d is viewed as a bag of words, which implies conditional independence of terms within the document. Using Eq. 3.1 and estimating priors as described in [33], the algorithm selected candidate category with the largest posterior probability.

Having thus selected candidate category, which is the end result of the *narrow-down* implementation in [33], the method proposed here reconsiders the original input hierarchy. Similarly to the approach taken in [33], our method associates each hierarchy node with a synthetic document, which is a concatenation of all training documents at all levels under the node. In addition, our method transforms the target hierarchy by adding a child node for each of the documents associated with nodes in the training set. This transformation is motivated by the desire to be able to compare hierarchy nodes as a function of their children irrespective of whether a node is a leaf or non-leaf node. It is important to note that the transformation applies only to documents associated with hierarchy nodes (at both leaf and non-leaf levels) in the training set, not hierarchy descriptions or labels. The transformation allows the approach to maintain a consistent view of both leaf and non-leaf nodes. Figure 3.2 exemplifies the transformation of the SOC hierarchy and associated training documents. It should be noted that, while the SOC training data [34] used in this study and exemplified in Figure 3.2 only contained documents at leaf nodes, in the general case, the algorithm would perform similar transformation for documents associated with non-leaf nodes if such training documents were available.

The proposed algorithm proceeds by considering changes in information content related to each node and the test document under classification as the hierarchy is traversed

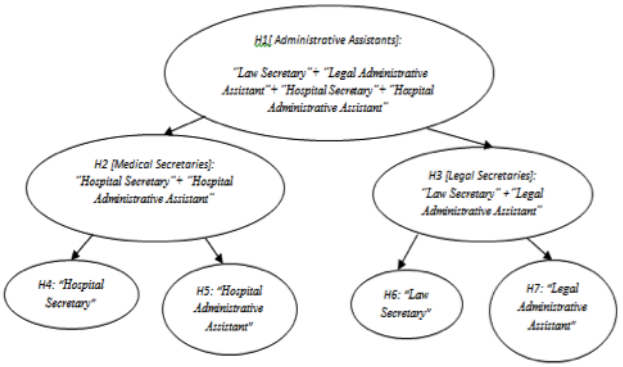


FIGURE 3.2: Hierarchy Transformation. Strings H1-7 represent node identifiers, text in square brackets (e.g: [Administrative Assistants]) represents node labels and quoted text represents documents associated with each node.

from each candidate category to the root. Information content, generally defined as $I(X = x_i) = -log(p(x_i))$, is a measure of the amount of information carried by random variable X when it assumes value x_i. In this context, we calculate the information content of observed values of random variables representing the events that the terms in the test document appear in the sub-tree under a given node. Formally, let $W = w_1, , w_k$ be a set of all possible terms and d be a test document. Further, let $T =< V, E, r >$ represent a taxonomy tree such that T is a directed acyclic graph, V is a set of vertices, E is a set of edges, r is the root node and each node in V may have at most one parent. Let a path through T to some node $h_j \in V$ be a sequence of nodes $L_j =< r, , h_j >$ such that every subsequent node is a child of the node that precedes it in the sequence L_j. Let $X = \{X_{11}, , X_{1|W|}, , X_{|V||W|}\}$ be a set of binary random variables such that $X_{ij} \in X$ represents the event of finding the jth term in W in some document associated with a node on a path that includes any decedent of the ith node in V excluding the ith node itself. The probability distribution $P(X_{ij} = b \in 0, 1)$ is estimated as

$$P(X_{ij} = b \in \{0, 1\}) = \frac{pf_{ijb}}{pf_{ij0} + pf_{ij1}} \tag{3.6}$$

where pf_{ij1} and pf_{ij0} are the numbers of paths in the training set where a document associated with any descendant of ith node in V contains and do not contain the jth term in W respectively.

Because a test document is viewed as a bag of words, variables $X_{i1}, , X_{i|W|}$ are assumed to be conditionally independent of each other in the document. Let S_d is the set of unique terms in document d, S_{h_j} is the set of unique terms in the synthetic document associated with node h_j and let $S_{d,h_j} = S_{h_j} \cap S_d$. The joint information content of positive realizations $X_{ij} \in X = 1$ of random variables in X for test document d and

some node h_j is

$$I_d^{h_j}(X) = \sum_{i=1}^{|S_(d,h_j)|} I(X_{ij} = 1) \tag{3.7}$$

where $I(X_{ij} = 1)$ is the information content of event $X_{ij} = 1$. The proposed algorithm uses the quantity $I_d^{h_j}(X)$ (Eq.) to evaluate the information content of terms in the test document for each of the nodes in the path from the candidate hierarchy to the root. Because the set of terms being evaluated for their information content value is restricted to the intersection of terms found the test document and the document associated with the node, probability of positive realization $p(X_{ij} = 1)$ is guaranteed to be greater than zero. Such guarantee is desirable as $log(0)$ is undefined.

To illustrate this evaluation, consider Figure 2 and imagine that the narrow-down algorithm [33] identified node H3: [Legal Secretaries] as the candidate category for a hypothetical input document Secretary.

$$\begin{aligned} I_{\text{“}Secretary\text{”}}^{H3} &= I(X_{\text{“}Secretary\text{”},H3} = 1) \\ &= -log(\frac{pf_{H3,\text{“}Secretary\text{”},b}}{pf_{H3,\text{“}Secretary\text{”},0} + pf_{H3,\text{“}Secretary\text{”},1}}) \\ &= -log(\frac{1}{2}) > 0 \end{aligned} \tag{3.8}$$

whereas

$$\begin{aligned} I_{\text{“}Secretary\text{”}}^{H1} &= I(X_{\text{“}Secretary\text{”},H1} = 1) \\ &= -log(\frac{pf_{H1,\text{“}Secretary\text{”},b}}{pf_{H1,\text{“}Secretary\text{”},0} + pf_{H1,\text{“}Secretary\text{”},1}}) \\ &= -log(\frac{4}{4}) = 0 \end{aligned} \tag{3.9}$$

In this example, the information content is lower at *H1* as the hierarchy is traversed from the candidate category *H3* towards the root. Since the information content is lower at *H1* than at *H3*, *H1* is selected among these two alternatives as the classification category for the test document.

3.3.1 Classification

Classification procedure computes joint information content of X for test document d and each of the nodes on the path from the candidate node to the root of the tree, inclusively. The element with the lowest value of joint information content is selected as the final classification for the input document. In information theory, the higher the information content the more uncertain the outcome. Because we use a set of random variables to represent the event of finding a term of the test document in some branches under a given node, intuitively, lower joint information content of the content of the test document would indicate higher predictability of the document.

It is possible, however, for more than one node on the path to have the same joint information content value. That is a problem since the algorithm must select only one of the nodes. To address this problem, a biasing parameter is introduced to allow implementations to skew the node selection decision based on domain-specific understanding of the target hierarchy:

$$I'^{h_j}_d(X) = \lambda_j + I^{h_j}_d(X) \tag{3.10}$$

In Eq. 3.10, parameter $\lambda_j < I^{h_j}_d(X)$ is a constant associated with jth level of the tree. The λ_j parameter is meant to serve as tiebreaker and bias the algorithm towards higher (or lower) levels of the hierarchy in those cases where more than one node on the path from the candidate node to the root contain the minimal amount of information content vis-a-vis the test document. To bias the algorithm towards higher (or lower) levels in a tree where 1st level represents the root, parameter λ_j for some level j could be chosen to be strictly greater (or strictly less) than λ_{j+1}. The additive nature of $I'^{h_j}_d(X)$ ensures that the algorithm has a chance to make a meaningful decision in those cases where $I^{h_j}_d(X) = 0$ for more than one node. In order to ensure that the parameter is only effective in the tiebreak situations, the parameter is computed in terms of lowest possible information content for a given taxonomy:

$$\lambda_j = \begin{cases} -j^{-1} * log(\frac{b^l-1}{b^l}) & \text{if } \textit{biasing towards higher levels} \\ -(l-j+1)^{-1} * log(\frac{b^l-1}{b^l}) & \text{if } \textit{biasing towards higher levels} \end{cases} \tag{3.11}$$

where b is the estimate of the number of children of each node of the target taxonomy tree, j is a one-based index and l is the height of the tree. Intuitively, the quantity

1. For a given input document, let d be the input document. Let $L_c =< r, , h_c >$ be the path from the root node r to candidate node h_c.
2. For each $h_i \in L_c$
   a. Calculate $I'^{h_j}_d(X)$ using Eq. 3.10
3. Assign the input document to node $h_{minima} \in P_c = argmin_{h \in P_c}(I'^{h_j}_d(X))$.

FIGURE 3.3: Classification Algorithm

$-log(\frac{(b^l-1)}{b^l})$ estimates the lowest potential information content (the smallest quantum) for a given tree that is strictly greater than zero. It quantifies a hypothetical situation of lowest possible information content for a given tree where some term occurs on all paths in the target hierarchy except one.

To exemplify the application of the parameter λ_j, consider the hierarchy in Figure 3.2. In the case of SOC matching, if a situation arises that for a given candidate category the minimal joint information content is the same for more than one node on the path from the candidate category to the root, biasing the algorithm towards higher nodes is intuitively appropriate. For the hierarchy in Figure 3.2 we construct parameters λ_1 and λ_2 for the top two levels:

$$\begin{aligned} \lambda_1 &= -1^{-1} * log(\frac{2^2-1}{2^2}) \\ \lambda_2 &= -2^{-1} * log(\frac{2^2-1}{2^2}) \end{aligned} \tag{3.12}$$

Thus computed, λ_1 and λ_2 parameters bias the algorithm to choose nodes in higher levels in the hierarchy. However, since these parameters are calculated as fractions of the estimate of the lowest potential non-zero information content, they are only significant in those cases where lowest information content is the same for more than one node. In other cases, Note that the λ parameters are only computed for the levels above and including the level of the candidate node. We do not consider levels below the candidate node as the algorithm is only concerned with nodes above or including the candidate category.

The classification algorithm is outlined in Figure 3.3:

29-0000 Healthcare Practitioners and Technical Occupations
29-1000 Health Diagnosing and Treating Practitioners
29-1060 Physicians and Surgeons
29-1062 Family and General Practitioners

FIGURE 3.4: Example of SOC hierarchy

3.4 Evaluation and Results

The proposed algorithm is motivated by the application in the area of public health and epidemiology where manually entered observation descriptors must be mapped to canonical representations such as the Standard Occupational Classifications (SOC) hierarchy before investigation may move forward. The proposed approach was evaluated using the Womens Health Initiative (WHI) observational study [35] data set. The data set consisted of 274,920 records, each one containing two unstructured text fields jobtitle and industry. Two algorithms were applied to the WHI data. The algorithm described in [33], which served as a precursor to the current approach, was used as the baseline. Performance improvements were demonstrated by comparing quality metrics of the baseline algorithm to those produced by the extension process described in this chapter.

The WHI data records were matched with the Standard Occupational Classification (SOC) hierarchy which is published by the Bureau of Labor Statistics. The SOC is a four level classification system with each level corresponding to progressively higher degree of aggregation. SOC levels include (in descending order) major group, minor group, broad occupation and detailed occupation. At the time of writing, the current revision published by Bureau of Labor Statistics contained 23 major groups subdivided into 97 minor groups containing 461 broad occupations which, in turn, contained 840 detailed occupations.

Each item in the SOC hierarchy is designated by a six-digit code of form ZZ-ZZZZ where Z is a digit. The first two (leftmost) digits designate the major group. The major group code is followed by a dash and the third digit, which represents the minor group. The fourth and fifth consecutive digits represent the broad occupation level of hierarchy. Finally, the sixth and last digit stands for the detailed occupation. Figure 3.4 below depicts an illustrative example of a SOC code hierarchy.

Bureau of Labor Statistics publishes several documents in order to communicate the SOC hierarchy and structure to the public. In addition to documents describing the structure and official definitions, it also provides a document containing a large number of text examples for each detailed and broad occupation groups.

The quality of the results produced by the matching process was evaluated in several ways. First, results were evaluated using manually coded job description strings using precision measure. Second, utility of the automatic classification was compared to that of the manual one via the use of physical demand measurements associated with each SOC code available through the O-Net [36] online system. The following sections describe the two evaluation methods in detail.

3.4.1 Manual Coding Evaluation

Two human coders (IB & YM) matched the free-text job descriptions of the first job reported to SOC codes. In order to improve productivity and focus on most important classifications, the team first grouped job description records by their job title component and proceeded to code more frequent job titles (all 75 that occurred in ¿100 records) as well as a sample or more rare ones (random sample with no replacement of 100 jobs that occurred with frequencies 2-99, and random sample with no replacement of 100 jobs that occurred with frequencies of 1). The resulting classification map contained 215 SOC matches established by consensus of the two coders. Table 3.2 illustrates the classification map by citing the top most frequent titles.

Using the classification map exemplified in Figure 3.4, precision values were calculated for the automatically matched codes for those records in the WHI data set where the case-folded jobtitle matched the case-folded job title entry in the manual classification map and the industry field was empty (allowing for fair comparison). Precision is a measure commonly used in information retrieval and machine learning to quantify quality of retrieval and classification results defined as $\frac{(\#correct\,matches)}{(\#correct\,matches+\#incorrectmatches)}$ (in this context, a correct match is identified when a SOC match produced by the algorithm is identical to that coded by human classifiers).

Job Title	**Count**	**SOC Code**
Teacher	3064	25-2000
Secretary	2394	43-6010
RN	954	29-1141
Bookkeeper	586	43-3031
Teaching	504	25-2000
Sales	472	41-0000
Office Manager	471	11-0000
Clerical	462	43-9061
Clerk	447	43-9061
Receptionist	403	43-4171

TABLE 3.2: Manually Matched Codes

To evaluate the approach using the precision measure described above the algorithm was applied to job title strings from the manually compiled classification map and results were compared for equality on three levels of SOC classification detailed occupation, broad occupation, and minor group.

Table 3.3 depicts overall precision results for detailed occupation, broad occupation, and minor group classifications

Level	**Precision Baseline**	**Precision Extended**	**Percent Improvement**
Detailed Occupation	0.2	0.51	155%
Broad Occupation	0.32	0.72	125%
Minor Group	0.84	0.85	1%

TABLE 3.3: Precision values for SOC gradations using existing (baseline) method and newly proposed approach (extended).

3.4.2 Evaluation of Utility of Classification in Assigning Physical Demand Score for Epidemiologic Analysis

The second evaluation phase employed analysis of the classification results using physical demand values associated with SOC codes and available through the O-Net service. The summary physical demand score for each SOC was calculated as the sum of assessments that were determined on a ordinal scale (1 to 5) by 'experts'. Physical demand scores, then, represented sum of the expert ratings for each SOC code. The specific assessments used in calculation of the physical demands score are the O-Net (version 16.0) variables describing abilities associated with each SOC (namely, the importance of "Dynamic Flexibility","Dynamic Strength","Explosive Strength","Extent Flexibility", "Gross Body Coordination","Gross Body Equilibrium","Manual Dexterity","Speed of Limb Movement", "Stamina","Static Strength","Trunk Strength"), contexts of SOC-coded jobs ("Exposed to High Places", "Spend Time Sitting", "Spend Time Standing", "Spend Time Climbing Ladders, Scaffolds, or Poles", "Spend Time Walking and Running", "Spend Time Kneeling, Crouching, Stooping, or Crawling", "Spend Time Keeping or Regaining Balance", "Spend Time Bending or Twisting the Body", "Spend Time Making Repetitive Motions") and activities associated with each SOC (namely the importance of "Handling and Moving Objects","Performing General Physical Activities"). Given the physical demand scores, the analysis proceeded to evaluate the matching algorithm in terms of these scores. Appendix I shows percent error for the top most frequent job descriptions in terms of their physical demand scores. Overall percent error for matches for titles was evaluated using a weighted average with weights corresponding to the number of records with particular title strings. The overall percent

error was 15.95% for the baseline algorithm and 3.8% for the algorithm presented here a 76% improvement. Further, the proposed method improved the baseline by 155% detailed occupation and 125% broad occupation precision measurements (reported in Table 3.3).

3.5 Discussion

The main contribution of the approach described above is its ability to detect and quantify generic content in test documents and allow documents lacking specificity to be classified at higher levels of the target hierarchy. This upward mobility of classification is an improvement over existing state-of-the-art hierarchical classification algorithms, which only aim at identifying the most likely branch of the hierarchy in terms of candidate categories located using a search routine.

Significant improvements in automated classification accuracy produced by the proposed approach may help alleviate much of the manual overhead associated with preliminary data processing steps in public health research. It is important to note that, since automatically derived classifications do not match manual classifications exactly (precision values are less than one), human involvement may still be necessary in those cases where better accuracy is desired. The proposed automated approach may be used in conjunction with the manual process significantly reducing the effort needed to associate unstructured data with well-defined structures such as the SOC hierarchy. Such reduction in the overhead of preliminary data processing may make it possible for public health researchers to consider studies that would otherwise be too costly or too time consuming to undertake.

As a consequence, the usefulness of the algorithm is limited by the need for human involvement. While the results produced by the algorithm are encouraging, the human component of the real-world application of the approach may significantly impair its scalability. For larger test sets, even a relatively small error rate would imply the need for significant and possibly prohibitive manual effort. This limitation of the approach is a concern and is a subject for future work.

3.6 Conclusions

In this chapter, we presented an extension to a successful hierarchical classification algorithm which allows for hierarchical generalizations to be made automatically. We

have shown that this type of approach results in over one hundred percent improvements in classification accuracy for the WHI data set and the Standard Occupational Classification (SOC) hierarchy.

It is important and encouraging to note that the approach proposed in this chapter has seen adoption in the recent work on , the proposed approach was adopted in a recent epidemiological study in [37]. The study's overall recommendation was that "automated translation of short narrative descriptions of jobs for exposure assessment is feasible in some settings and essential for large cohorts, especially if combined with manual coding to both assess reliability of coding and to further refine the coding algorithm" [37].

Chapter 4

Hierarchical Text Mining for Web Summaries

The preceding chapter 3 outlined an approach for automatic association of unstructured text strings with a well-defined, standard taxonomy. While this was shown to be helpful in real-world applications exemplified in [37]. However, the approach has not been shown to be fully applicable for other types of short and noisy data, such as the microblogs, which are the overall goal of this thesis. Further, while the algorithms discussed in previous chapters were shown to be useful in matching unstructured records into a given taxonomy, they do not expose a way to infer statistical summaries or views of content. As such views may be helpful for individuals aiming to grasp the nature of the content, this chapter presents a different approach that automatically discovers statistical views from unstructured text data and predefined taxonomy.

4.1 Introduction

Recent developments in information technology have resulted in volumes of new Web content being continuously added to the World Wide Web. Millions of new websites and web documents are being made available over the Internet making organizing, filtering and classifying this ever-increasing amount of content a daunting task.

As the amount of web content increases and becomes harder and harder to grasp, organizations emerge that take it upon themselves to try to organize and classify public web sites. One such organization is the Open Directory project (also known as DMOZ). The project aims to organize web content by compiling a comprehensive directory of public web sites available on the Web. This Web directory compilation is accomplished

through the hard work of thousands of human volunteers who inspect and manually map websites based on their content into a well-defined ontology [38].

While the efforts of the DMOZ project certainly go a long way towards developing an organized repository of worlds Web data, DMOZ ontology elements provide no information beyond their textual labels as to the content and general meaning of associated documents. That is, even though much work has been done by the Open Directory project towards organizing, classifying and providing navigational aids for collections of Web documents, the project makes no attempt to extract other information about documents in those collections including path-level and statistical views of content. Eliciting such additional information may facilitate tasks such as browsing, searching, and assessing document similarity [39].

Therefore, in this work, we attempt to leverage the manually created ordered ontology structures published by the Open Directory project to model the content of underlying document collections. We aim to produce a view of the content which would enable individuals to quickly grasp underlying themes of documents associated with ontology nodes. We accomplish this aim by relying on topic modeling techniques and formalisms. We develop an approach for constructing a statistical view of each ontology node by associating nodes with topics, which are customarily viewed in topic modeling as probability distributions over a fixed vocabulary. Our technique estimates distribution parameters as word-multinomials and uses these multinomials to produce sorted lists of vocabulary terms for all nodes in the ontology. Qualitative evaluation of the sorted lists produced by our technique suggests that top most probable terms, as specified by the corresponding word-multinomial parameters, are indicative of the underlying general theme of Web documents associated with corresponding ontology nodes.

We propose a new probabilistic generative model based on a Labeled-LDA [40] approach, which is a supervised variant of the well-known LDA [41] model. The new Tree-Labeled LDA model takes advantage of hierarchical nature of the DMOZ ontology and jointly models word and ontology node assignments as a generative process.

We evaluate our approach quantitatively by comparing predictive power of resulting topic models with that of topic models produced by other state-of-the-art algorithms, in terms of perplexity, for held-out data. We show that, for datasets used in the study, the new tLLDA model outperforms other state-of-the-art Labeled LDA and Hierarchically Labeled LDA topic modeling approaches in terms of perplexity. As no quantitative way are available to assert the relevance of topic models to underlying content [42], we conduct qualitative evaluation of the resulting topic models as compared to topic models produced by Labeled LDA and Hierarchically Labeled LDA and conclude that

topic models produced by the new tLLDA qualitatively more semantically indicative of the underlying content in the view of the authors.

The rest of the chapter is organized as follows. Section 4.2 presents review of notable related works. In section 4.3, we introduce the generative Tree Labeled LDA (tLLDA) model in detail. Section 4.4 discusses procedures for estimate distribution parameters from data. In section 4.5, we discuss experimental setup and evaluate language models produced by the tLLDA approach as compared with other algorithms. In sections 4.7 and 4.8 we discuss conclusions and outline subsequent work.

4.2 Related Works

The now classical work on Latent Dirichlet Allocation (LDA) [41]] has provided an extensible modular framework for many topic modeling approaches. In LDA, the basic idea is that documents are represented as random mixtures over latent topics with each topic characterized by a distribution over words [41]. The approach championed by LDA has enjoyed popularity in the topic modeling community with many works extending and generalizing on the basic principle. While Latent Dirichlet Allocation has served as basis for many approaches [39], it is fully unsupervised and may not be the best choice when the goal is prediction. That is, for instance, in a system concerned with movie ratings intuitively good predictive topics could differentiate between excellent, terrible and average ignoring the genre. Unsupervised machinery of the basic LDA, however, may estimate topics that correspond to genres if that is the intrinsic structure of the corpus [39].

Fortunately, the extensible nature of LDA paves the way for derivative approaches that take supervision into account. One such approach is the Supervised LDA model (sLDA) [39], which extends LDA by adding a response variable associated with each document. This variable is usually associated with the supervisory labels for documents, such as the film rating adjectives in the above example. The sLDA model jointly models documents and responses with the goal of finding latent topics that will best predict responses for test data [39]. The response values come from a normal linear model, which covariates in the sLDA model with empirical frequencies of topics in documents [39].

A further refinement on the Supervised LDA model the Hierarchically Supervised LDA (HSLDA) [43] extends sLDA to take advantage of hierarchical supervision. The HSLDA model is based on the intuition that hierarchical context of labels provides valuable information about labeling. As in sLDA, HSLDA jointly models documents and responses by drawing response variable realizations from a Normal distribution, but unlike sLDA

it generates label responses using a hierarchy of conditionally dependent probit regressors [43]. In the joint modeling of each document, both empirical topic distribution and whether or not the parent label is applied to the document determine whether or not a label is to be applied. The HSLDA model views word-multinomials (topics) as global constructs and links them to hierarchy nodes through per-label topic distributions. This makes HSLDA output difficult to interpret as the global topics do not directly correspond to nodes.

While Supervised LDA and Hierarchically Supervised LDA have been shown to work well in some applications, they have the limitation of allowing only a single label to be applied to a document and are thus not applicable to document collections where multiple labels can be assigned to texts [40]. Therefore, a radically different way of providing supervisory input to topic modeling was developed. Named Labeled LDA (L-LDA), this model aimed at joining the multi-label supervision frequently found in modern text databases with word-assignment disambiguation of LDA family of models [40]. In L-LDA, each unique label is viewed as a topic and the goal of the model is to restrict the generative process to operate over a subset of topics, thus allowing for each document to be supervised.

Similarly to L-LDA, Hierarchically Labeled LDA (or hlLDA) [43] constricts the general LDA model to operate over a subset of label-bound topics. While L-LDA is a general model for corpora where documents may be associated with multiple topics that may or may not be related to each other in any way, hlLDA considers each document to be associated with a set of topics which corresponds to the set of nodes on the hierarchical path for each document. The hlLDA model relies on some of the formalisms described in yet another LDA successor model called Hierarchical LDA (hLDA) [44], which is an unsupervised generative model which infers hidden hierarchical structures from data. The hlLDA variant extends the hLDA approach by assuming that the hierarchical structure which is hidden from hLDA is known and restricts topics accordingly. Unfortunately, the hlLDA model is limited as it only considers supervision from observed hierarchy path for each document and does not make use of the hierarchy structure as a whole.

4.3 Model

In this section we introduce the Tree Labeled LDA (tLLDA) which is a generative probabilistic model that describes the process of generating a document collection where each document is associated with a distribution over hierarchy nodes. It improves upon hlLDA by jointly modeling word and node label assignments, which allows it to take the hierarchy structure as a whole into consideration. Further, the proposed tLLDA model

estimates a single word-multinomial for each node of the target ontology, which allows for an easier interpretability when compared with the HSLDA model.

The tLLDA model aims to incorporate both the multi-label supervision and supervision derived from the structure of the target hierarchy. Similar to other topic modeling techniques [43], it adopts the mixed membership formalism [41] where a document is thought of as a mixture over a set of word-multinomials [45]. The tLLDA approach marries the multi-label supervision of hLLDA with hierarchical supervision by jointly modeling word and label assignment generation. Also, unlike HSLDA which generates label responses using conditional hierarchy of probit regressors [43] assuming a Normal distribution, tLLDA draws a path through the hierarchy for each topic directly from a distribution parameterized by a global vector of multinomial parameters without assuming any underlying distribution shape.

Borrowing notation from [40], let each document d be represented by a set of word indices $w^{(d)} = \{w_1, , w_(N_d)\}$ where each $w_i \in \{1, , |W|\}$, W is the vocabulary and N_d is the document length. Let hierarchy $H = (V, E, c_r)$ be a tree which is a directed acyclic graph with known structure where $V = \{c_i, , c_(N_v)\}$ is the set of vertices (or nodes) of size N_v, E is the set of edges, $c_r \in V$ is the root node and each node $c_i \in V$ may have at most one parent. Then, let K be the number total number of topics equal to the size of the set V. We note that, since the number of topics equals the number of vertices in the hierarchy, topics and vertices (nodes) may be used interchangeably in this context. Let a path to a node $c_j \in V$ be the list $path_j = (c_r, , c_j)$ such that $c_i \in path_j \in V$ and each subsequent node in the list $path_j$ is a child of its predecessor. Then, for each node $c_i \in V$ let $descendants_i$ be a set a set of descendants of c_i. S_k represent a subtree rooted at node k.

The generative process of the algorithm is outlined in Figure 4.1

$$\lambda_j = \begin{cases} 1 & \text{if } c_i \in S_k \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

There, steps 1 and 11 where the multinomial topic distributions B_k over vocabulary for each topic $k \in \{1, \ldots, K\}$ conditioned on the Dirichlet prior η are drawn remain identical to the standard LDA, L-LDA and hlLDA.

To use the hierarchy structure as a supervising agent for the generative process we deterministically construct vector $\Pi^k = (k'_1, \ldots, k'_K)$ such that $k'_i \in \Pi^k \in \{0, 1\}$ for each topic k in steps 21-2(1)1.

1. For each topic $k \in \{1, \ldots, K\}$:
   1. Generate $B_k = (B_{k,1}, \ldots, B_{k,V})^T \sim Dir(\cdot\eta)$
2. For each topic $k \in \{1, \ldots, K\}$:
   1. For each topic $k' \in \{1, \ldots, K\}$:
      1. Deterministically set $\Pi^k_{k'} \in \{0, 1\}$ using Eq.4.1
   2. Generate $\Pi^k = R^k \times \pi$
   3. Generate $P_k = \{P_{k,1}, \ldots, P_{k,C_k}\}^T \sim Dir(\cdot|\pi^k)$
3. For each document d:
   1. For each topic $k \in \{1, , K\}$:
      1. Generate $\Lambda^d_k \in \{0, 1\} \sim Binom(\cdot|\phi_k)$
   2. Generate $\alpha^d = L^d \times \alpha$
   3. Generate $\theta^d = (\theta_{l_1}, \ldots, \theta_{M_d})^T \sim Dir(\cdot|a^d)$
      1. For each word $i \in \{1, , N_d\}$:
         1. Draw $z_i \in \{\lambda^d_1, , \lambda^d_{|M_d|}\} \sim Mult(\cdot|\theta^d)$
         2. Draw $w_i \in \{1, \ldots, V\} \sim Mult(\cdot|B_{z_i})$
         3. Draw $h_i \in \{1, , K\} \sim Mult(\cdot|P_{z_i})$

FIGURE 4.1: TLLDA generative algorithm

Because the hierarchy structure is known *apriori* and is fixed for all documents, this deterministic step does not jeopardize the generative nature of the approach. Having thus generated vector Π^k we define a node-specific matrix R^k over $C_k \times K$ for each node where $C_k = |\pi^k = \{k|\Pi^k_k = 1\}|$; for each row $i \in \{1, \ldots, C_k\}$ and column $j \in \{1, \ldots, K\}$ according to Eq. 4.2

$$R^k_{ij} = \begin{cases} 1 & \text{if } p^k_i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

The matrix R^k_{ij} is used to project parameter vector of the Dirichlet prior $\pi = (\pi_1, \ldots, \pi_K)^T$ to lower dimensional vector π^k in step 22 according to Eq. 4.3

$$\pi^k = R^k \times \pi = (\pi_{p^k_1}, \ldots, \pi_{p^k_{C_k}}) \tag{4.3}$$

In step 22, path assignment proportion vector P_k is drawn for each topic with parameter vector π^k. As the parameter vector π^k is constructed by projecting K-dimensional vector onto the lower dimensional space defined by ks subtree, it is in this step where the structure of the hierarchy is included into the supervision since parameter vector π^k is constructed by the use of S_k, the makeup of vector P_k depends on where the node corresponding to topic k is in the target hierarchy in relation to other nodes.

Then, for K number of topics, let $\Lambda^d = (l_1, \ldots, l_K)$ be the list of binary topic presence/absence indicators for document d such that $l_k \; in\{0, 1\}$. As in L-LDA, vector Λ^d is generated in steps 31-3(1)1 by using a binomial distribution for each topic k with prior ϕ_k. With that, a document-specific label projection matrix L^d over $M_d \times K$ is defined for each document where $M_d = |\lambda^d = \{k|\Lambda^d_k = 1\}|$; for each row $i \in \{1, , M_d\}$ and column $j \in \{1, , K\}$ according to Eq. 4.4.

$$L^d_{ij} = \begin{cases} 1 & \text{if } \lambda^d_i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

The matrix L^d is used to project the parameter vector of the Dirichlet prior $\alpha = (\alpha_1, \ldots, \alpha_K)^T$ to lower dimensional vector α^d according to Eq. 4.5.

$$\alpha^d = L^d \times \alpha = (\alpha_{\lambda^d_1}, , \alpha_{\lambda_{M^d_d}}) \tag{4.5}$$

In step 33, θ^d is drawn by parameterizing a Dirichlet distribution with α^d computed in step 32. Then, to generate a word, topic k is sampled from a distribution parameterized by θ^d and a word is sampled from distribution over words parameterized by the word proportion vector B_k.

Unlike other algorithms that repeat the process at this point, tLLDA draws a path assignment $h_i \in S_k$ from a distribution parameterized by vector P_k where k is the topic drawn in step 23. As the vector is projected onto the lower dimensional space defined by ks subtree, it is in this step where the structure of the hierarchy is included into the supervision – makeup of S_k and consequently the values of vector P_k depends on where the kth node is in the target hierarchy in relation to other nodes.

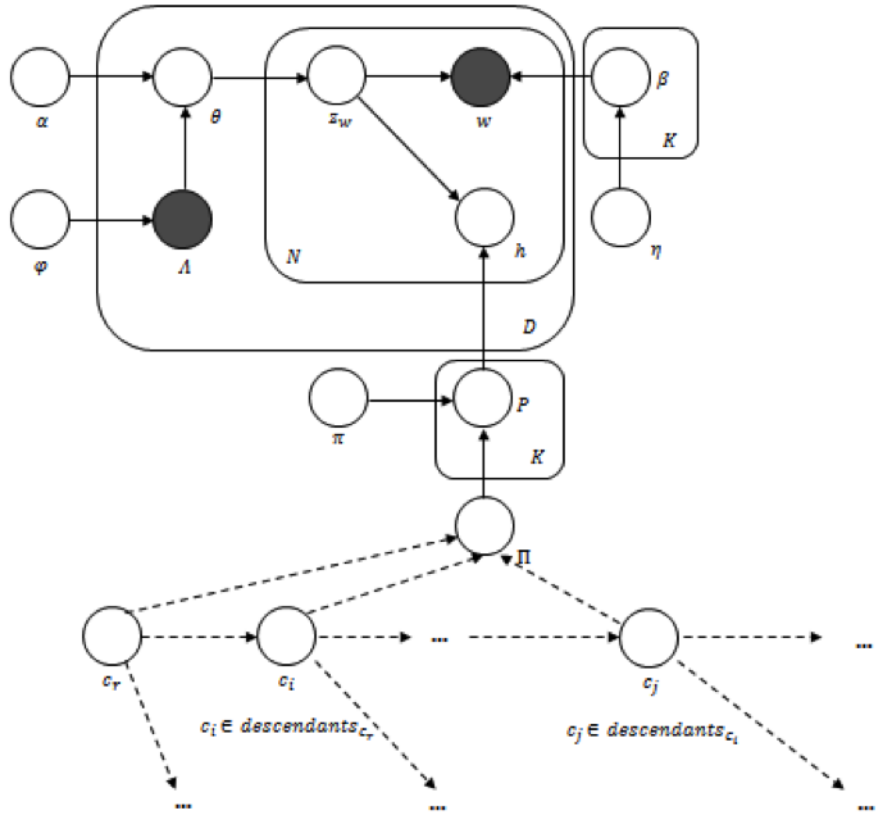


FIGURE 4.2: TLLDA graphical model showing the plate diagram with solid lines representing probabilistic links and dashed lines representing deterministic relationships. The shaded circles represent observed nodes whereas un-shaded nodes are hidden.

4.4 Parameter Estimation

We used Gibbs sampling to estimate topic-word distribution parameter θ^d. Compared to other parameter estimation methods, Gibbs sampling yields a relatively simple and computationally efficient algorithm [42].

Sampling equation for a topic for document d and path leading to c_d (notation outlined in Table 4.1):

η, α, π	Dirichlet hyperparameters
$n^{w_i}_{-i,j}$	The number of times word w_i assigned to topic j not counting the current instance
$n^{\cdot}_{-i,j}$	Number of times any word is assigned to topic j excluding the current instance
$n^{d}_{-i,j}$	The number of times topic j is assigned to document d excluding the current instance
$n^{d}_{-i,\cdot}$	Number of times any topic is assigned to document d
$p^{k'}_{-i,j}$	$n^{d}_{-i,\cdot}$ if $j \in path_{k'}$; 0 otherwise
$p^{\cdot}_{-i,j}$	$\sum_{m=1}^{S_j} n^{\cdot}_{-i,m}$
c_d	Observed node assignment for document d; $c_d \in \{1, , K\}$

TABLE 4.1: Notation

```
<Topic r:id="Top/Arts/Animation/Anime/Fandom/A">
    <ExternalPage about="http://www.atanime.net/">
    <d:Title>At Anime.Net</d:Title>
    <d:Description>Contains themes and wallpapers.</d:Description>
⋮
<Topic r:id="Top/Arts/Animation/Anime/Fandom/B">
    <ExternalPage about="http://members.tripod.com/Okami_N_Akume/index.html">
    <d:Title>Beyond Eternal</d:Title>
    <d:Description>Sailor Moon and Final Fantasy information and fan art.</d:Description>
⋮
```

FIGURE 4.3: Example snippet of the World DMOZ dataset

After a predetermined number of iterations of the sampling process based on distributions estimated using the above equation, parameters can be estimated for any single sample as using the following equations:

$$\theta_k^d = \frac{n_k^d + \alpha}{n_\cdot^d + |path_{c_d}|\alpha}; B_k^{w_i} = \frac{n_k^{w_t} + \eta}{n_k^\cdot + |W\eta|}; P_k^{k'} = \frac{p_k^{k'} + \pi}{p_k^\cdot + |S_k|\pi} \tag{4.6}$$

4.5 Experiments and Results

4.5.1 Experimental Data

We tested the tLLDA on the Web summary dataset provided by the Open Directory project. The Open Directory project publishes four distinct data dump pairs (structure file and content file). These are World, Kids and Teens, Adult and AOL. These archives constitute voluminous data sets consisting of over two million entries and published as XML-like RDF files. The DMOZ Web directory is organized in a well-structured ontology with lower-level topics increasing in specificity and lateral links connecting related topics. The ontology structure is published in a separate RDF documents devoid of review content. Review content RDF files are made up of sections corresponding to ontology identifiers (topics). Each section contains a set of URL references to websites associated with website title strings sequestered directly from each website markup as well as short description strings written by Open Directory volunteer reviewers. Both the title and description strings are relatively short with the average of 4.2 and 18.1 words per string respectively.

The World RDF download contains ontology structure and corresponding reviews for all web sites reviewed by the project excluding those found in Kids and Teens, Adult and AOL repositories. The Kids and Teens download contains reviews of web sites related to children and teenagers such as reviews of cartoons and primary school activities. Adult repository contains reviews of adult material. In our experiments, we excluded

1. Set N to be the total number of records in a dataset
2. Set $S = \{1000, 2000, 3000, 4000, 5000, 6000\}$
3. While $S \neq \emptyset$
    1. Randomly draw $s \in S$ from uniform distribution over the set S
    2. Randomly draw a burn-in number n from a uniform distribution over a set $\{1, \ldots, N - s\}$
    3. For each record in a depth-first traversal of the hierarchy structure
        1. If the index i of the record $i > n$ *and* $i < (n + s)$, then
            1. Randomly draw a number r from a uniform distribution over discrete set $\{1, \ldots, s/10\}$
            2. If $r = 1$, add record to test set associates with $s \in S$
            3. Otherwise, add record to the training set associated with $s \in S$
    4. Set $S = S \backslash s$

FIGURE 4.4: Algorithm for extracting the train data window

the AOL repository as the latest file published at [9] contained no review content. We also excluded the Adult section from our experiments for ethical reasons and tested our approach on the World and the Kids and Teens datasets. As both the World and Kids and Teens repositories were large (2.1 million records and 26,000 respectively) and because of available hardware and time constraints we tested our approach using a set of smaller subsets of records extracted from each repository. For our experiments, we used the English language portions of the World and the Kids and Teens portion of the DMOZ datasets. As preprocessing steps, each raw record, title and description string were extracted and concatenated into a single document. Resulting strings where case-folded and tokenized using simple tokenization rules followed by non-letter characters (numbers and punctuation) removal. No stemming or lemmatization was applied. Following the process in related works [8], vocabulary was extracted by a single pass through the data and documents were regenerated by replacing English terms with numerical identifiers of terms in the vocabulary.

Because of the sparse nature of data sets which are made up of relatively small number (average of 5.4 documents per hierarchy node) of short review documents and in order to preserve the underlying relationships within the hierarchy, we extracted six fixed size data windows of various sizes from each of the two data sets. Further, we reserved 10% of the data in each of the data windows for testing by applying simple random sampling without replacement. The train data windows and test records were sequestered according to the procedure in Figure 4.4.

It is apparent from the above algorithm description that the maximum number of records used for testing the approach was limited to 6000 elements. This number was chosen as it was empirically determined to be the largest number of records that the available hardware was able to process in one twenty-four hour period for each algorithm tested.

4.5.2 Comparison Models

We compared the predictive power of language models produced by our approach to four related models against the Open Directory (DMOZ) dataset. The four comparison models included the Hierarchically Labeled LDA and the Labeled LDA, Hierarchically Supervised LDA and Supervised LDA. We considered two distinct comparison approaches as evaluation criteria. We used perplexity to compare tLLDA performance with that of hlLDA and L-LDA as perplexity is a common way to compare language models. To compare tLLDA predictive power with that of sLDA and HSLDA we used multilabel classification precision as evaluation criteria. We chose to use multilabel classification as opposed to perplexity for sLDA and HSLDA because sLDA-type approaches associate supervisory topics with distributions over language models rather than with single language model making evaluation in terms of perplexity not feasible.

4.5.3 Perplexity Evaluation

We compare language models produced by our approach to those learned by hlLDA and L-LDA as the output of these models is lets itself to per-topic language model comparison. To compare language models, we used perplexity measure over held-out subset of data $W = \{w_1, , w_n\}$ given language model M and the training data [45] calculating perplexity via Equation 4.7.

$$perp_M(\overline{W}) = exp(-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\overline{W}|}\sum_{j=1}^{|\overline{w}_i|} log(p_m(\overline{w}_{ij}))) \tag{4.7}$$

where $n = |\overline{W}|$, $\overline{w} \in \overline{W}$, $\overline{w}_{ij}$ is the jth term in the ith string in the held-out collection and $p_M(w \in \overline{w})$ is the probability of term w as per the learned language model M.

The tLLDA approach outperformed L-LDA and hlLDA algorithms in terms of perplexity. Tables 4.2 and 4.3 summarize experimental results for World and Kids and Teens data subsets respectively in terms of perplexity values. We set $\alpha = 0.5$, $\eta = 0.03$ and $\pi = 1$

1. For each $k \in \{1, , K\}$
   1. Let $s_k = \{l | l \in path_k\}$
2. Let $T = \{t_i, \ldots, t_n\}$ be a set where $t_i \in T$ is a document and n is the number of test documents
3. For each tT
   1. For each $k \in \{1, \ldots, K\}$
      1. Estimate probability $p(s_k|t)$ using approach-specific heuristic
4. For each $t \in T$
   1. Let document t be classified as $argmax_{m \in \{1,\ldots,K\}} p(s_m|t)$

for all experiments. We set the number of iterations to 1000 as the algorithm appeared to converge past that number of iterations.

	1000	**2000**	**3000**	**4000**	**5000**	**6000**
L-LDA	2642.63	3952.49	4661.74	6016.89	6740.44	8142.88
hlLDA	3159.34	4598	5714.75	7046.47	8264.21	9787.72
tLLDA	**690.25**	**795.99**	**779.99**	**877.47**	**1077.13**	**1356.14**

TABLE 4.2: World dataset perplexity values for L-LDA, hlLDA and tLLDA (rows) and 1000-6000 record data windows (columns)

	1000	**2000**	**3000**	**4000**	**5000**	**6000**
L-LDA	4170.66	4958.17	6232.11	7450.27	8226.14	8976.4
hlLDA	4081.14	6247.47	7767.5	8809.1	9421.62	10035.43
tLLDA	**838.26**	**965.94**	**1165.07**	**1297.19**	**1400.46**	**1473.13**

TABLE 4.3: Kids and Teens dataset perplexity values for L-LDA, hlLDA and tLLDA (rows) and 1000-6000 record data windows (columns)

4.5.4 Multilabel Classification Evaluation

Since sLDA-type approaches associate supervisory topics with distributions over language models rather than with single language model per topic as in the case of hlLDA and L-LDA, it is difficult to meaningfully compare language models of tLLDA with those of sLDA and HSLDA in terms of perplexity. Therefore, we opt for a different measure and compare performance of tLLDA to that of sLDA and HSLDA by applying multilabel classification, which is a task of predicting the set of labels appropriate for each document given a training set of documents with multiple labels.

The general procedure for multilabel classification used in this paper is outlined in Figure **??**

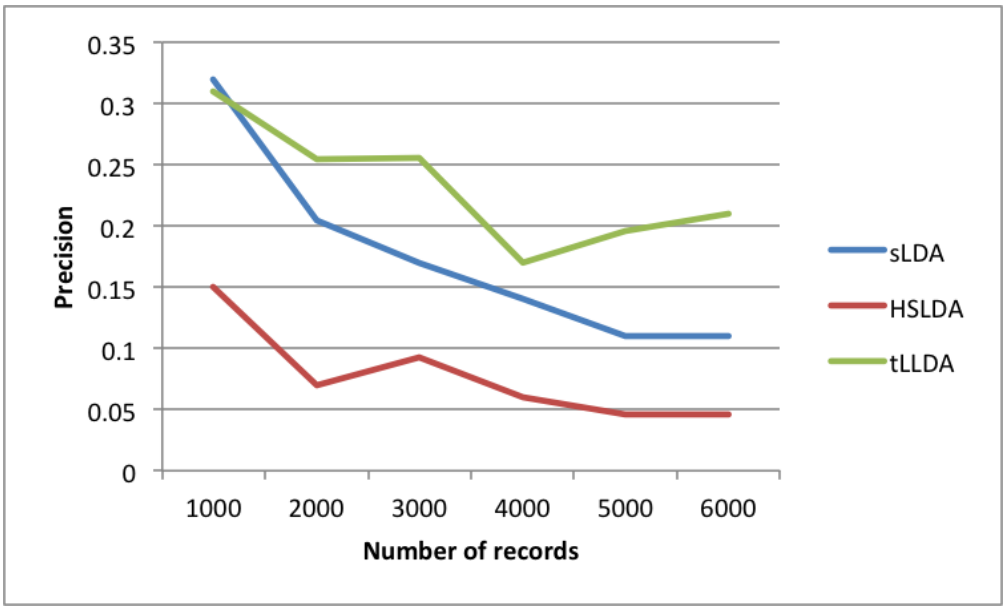


FIGURE 4.5: Precision results for each test data window for the World dataset

In this study, the classification algorithm remained the same for tLLDA, sLDA and HSLDA with the only variation being the posterior probability approximation procedure. Since the goal of the classification task is to try and guess the set of labels (a path) for each test document, document-topic approximation routine used during testing must differ from that which was used during model training for topic modeling approaches that take supervision into account. Therefore, to estimate document-topic probabilities, all tested algorithms estimated label-level and global parameters during training and used those estimates in unsupervised LDA inference to estimate document-level distributions.

We gauge the results in terms of precision which is defined as $\frac{\text{number of test documents classified correctly}}{\text{number of test documents}}$. Figures 4a and 4b show precision results for tLLDA, sLDA and HSLDA for each sample data window for World and Kids and Teens datasets respectively. Fugures 4.5 and 4.6 show precision results for tLLDA, sLDA and HSLDA for each of the test data windows for the World and the Kids and Teens datasets respectively

4.5.5 Topic Visualization

Recalling that the goal of this study was to produce a view of the content which would enable individuals to quickly grasp the underlying theme of documents associated with each ontology node and realizing that no clear means existed for quantifying the quality of topic multinomials, we evaluated the topics discovered by our model by examining the top words assigned to each topic. We observed that the top word assignments produce by the tLLDA model appear semantically more meaningful as compared to those produced by hlLDA and L-LDA as the language models produced by the later algorithms seem to

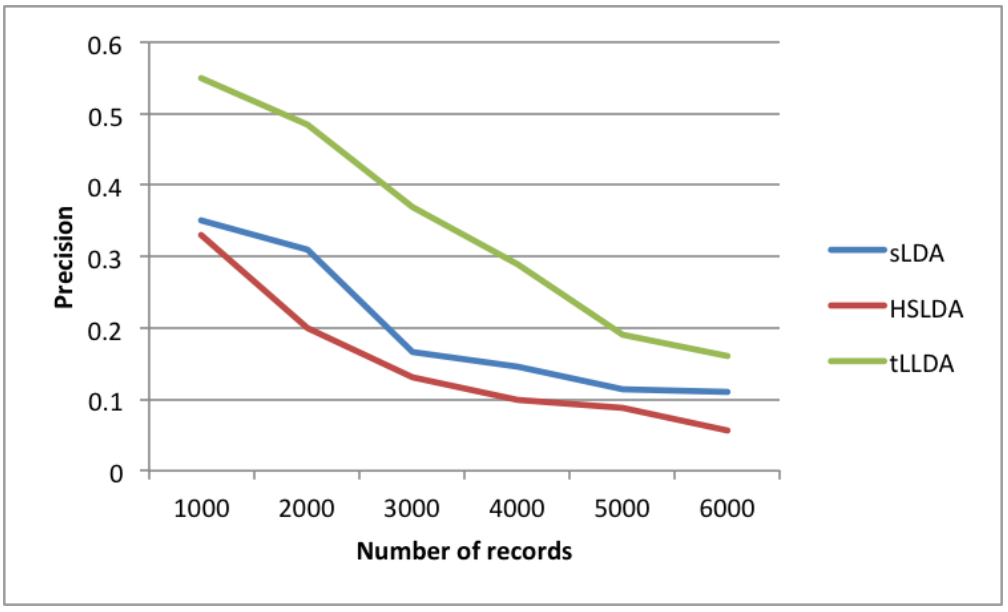


FIGURE 4.6: Precision results for each test data window for the Kids and Teens dataset

favor proper nouns for language models associated with lower-level hierarchy nodes and marginalize other terms that add semantic consistency to evaluation. Table 4.4 contains example topic distributions for the top two evaluated modeling approaches.

4.6 Discussion

Similar to L-LDA, one of the advantages of tLLDA is the document-specific topic mixture . In the context of Web documents, the topic mixture can be inferred for each new web page and since each topic is associated with a node in the target ontology, insight as to the subject matter of each document can be gained by evaluating the topic proportions. Considering the sheer volume of documents available on the Internet, such insight may be instrumental in helping organizations such as the Open Directory project in their important tasks of digesting the Web content and presenting it to the public in an accessible way.

The node-specific topic proportions learned by the tLLDA approach may be beneficial to Web classification efforts in several ways. As topic proportions are learned from the underlying collections of documents, it may be possible to consider words in terms of their probability vis--vis the corresponding node-level language model and consider the structure of the target hierarchy. As target ontologies, such as the one used by DMOZ, are manually compiled, they may contain omissions or overgeneralization. Such omissions and overgeneralizations may become prominent when viewed through the prism of their learned language models.

	Composers	**Composers /Classical**	**Composers/ Classical /Beethoven, _Ludwig_van**
tLLDA	and with includes for a his samples biography composers songs audio of the biographies music who children works illustrated lives antonio computer classical brief lyrics vivaldi tchaikovsky pytor mozart berlin johan parlor	of composer the on work classical grieg edvard in a information his profile piano life famous minor death times concerto	biography beethoven van ludwig and brief works of a includes list
L-LDA	biography brief composer the of with for key pronunciation factmonster fact monster italian thinkquest franz austrian russian grave english antonio german french peter haydn vivaldi tchaikovsky beethoven wolfgang bach ludwig	mozart amadeus life wolfgang work strings mozarts mountains grieg edvard and form profile piano minor interesting moguls	van beethoven ludwig includes compositions introduction comprehensive discussion carole awesome

TABLE 4.4: Sample topic visualizations for each evaluated algorithm (rows) and several hierarchy levels (columns). Highlighted terms indicate words that appeared semantically indicative of the content theme during qualitative evaluation by the authors.

To exemplify, consider the sample cited in Table 4. There, the set of terms "biography beethoven van ludwig and brief works of a includes list" is more probable for the hierarchy node with the path "Composers Classical Beethoven, Ludwig van" as per the node-specific language model learned by the tLLDA approach. Examining this list of most probable terms may suggest that the underlying collection of documents related to the famous composer may be further partitioned into two more specific subcategories one containing documents related to the biography of the composer and the other related to his works.

4.7 Conclusions

In this chapter, we proposed a new topic modeling approach of Web summaries using a popular Web ontology. This approach took advantage of the hierarchical structure of the ontology to improve predictive power of resulting topic models. While we focused on the Web summary data provided by the Open Directory project, the topic modeling approach introduced here can be easily adopted to any hierarchically organized content. This type of model can be useful in identifying key notions in large collections of text.

4.8 Next Steps

In this work, we took advantage of tree-like hierarchical structure exhibited by the Open Directory ontology. In addition to the hierarchical relationships, however, the ontology also provides lateral links between related nodes that often breach hierarchical boundaries of parent-child relationships. Understanding how to take advantage of these links to further improve predictive power of topic models is one area of future research.

As much of our evaluation was hindered by the computational complexity of parameter estimation algorithms, in future works we will attempt to leverage hierarchical structures to reduce time required to estimate parameters for the proposed model for larger datasets by distributing the parameter estimation process. The need for distributed topic modeling has been repeatedly expressed by researchers and much work has already been done in this area. In future work, we will attempt to build on earlier works on distributed topic modeling algorithms and use intuitions and findings of this work to improve performance of the tLLDA parameter estimation procedure.

Chapter 5

Scalable Hierarchical Topic Mining of Social Streams

In this chapter, I apply lessons learned from previous effort that focused on occupational health data to the study of topic modeling in continuous social media streams. To try and overcome the narrow focus of previously discussed approaches, I propose a new generative probabilistic model called Hash-Based Stream LDA (HS-LDA), which is a generalization of the popular LDA approach. The model differs from LDA in that it exposes facilities to include in- ter-document similarity in topic modeling. The corresponding inference algorithm outlined in the paper relies on efficient estimation of document similarity with Locality Sensitive Hashing to retain the knowledge of past social discourse in a scalable way. The historical knowledge of previous messages is used in inference to improve quality of topic discovery. Performance of the new algorithm was evaluated against classical LDA approach as well as the stream- oriented On-line LDA and SparseLDA using data sets collected from the Twitter microblog system and an IRC chat community. Experimental results showed that HS-LDA outperformed other techniques by more than 1% for the Twitter dataset and by 21% for the IRC data in terms of average perplexity.

5.1 Introduction

The work presented in this chapter is motivated by the problem of topic discovery in social media. We recognize that topic discovery systems for online social discourse need to address a set of challenges associated with the scale of modern social media outlets such as Twitter, chat systems and others. To be useful, these systems must operate

continuously for extended periods of time, as social conversations do not stop, produce output in a timely fashion to remain relevant and ensure high quality of output.

Commonly used data mining techniques handle the problem of social stream topic discovery by applying batching heuristics to process the never-ending stream of messages. Since retaining all messages is not feasible in practice, current topic modeling approaches improve quality of topic discovery by retaining globally applicable statistics such as topic-word counters, but fail to take advantage of document-level information as no technique has existed so far to retain such information in a scalable and meaningful way.

Therefore, in this work we propose a new generative probabilistic model called Hash-based Stream LDA (HS-LDA), which is a generalization of the popular Latent Dirichlet Allocations (LDA) [41]. The model improves upon previous works by introducing a theoretical framework that makes it possible to retain the knowledge of historical stream messages in a scalable way and use this knowledge to improve the quality of topic discovery in social streams. Further, an efficient inference mechanism for the HS-LDA model is outlined, which makes use of the scalable hashing algorithm called Locality Sensitive Hashing (LSH) [46]. We show that the HS-LDA model and the associated inference algorithm are well suited for topic discovery in streams by comparing the predictive power of the topic models inferred by HS-LDA with that of topics learned by applying the classical LDA, On-line LDA [47] and SparseLDA [48] approaches to stream data. Evaluation was performed using data collected from the Twitter microblog site and an IRC chat system. Our experiments showed that HS-LDA outperformed other techniques by more than 12% for the Twitter dataset and by 21% for the IRC data in terms of average perplexity.

This chapter is organized as follows. In section 5.2, current state of the art of topic modeling and stream mining is discussed. Section 5.3 introduces the HS-LDA model, outlines an efficient inference algorithm and discusses its application to stream data. In section 5.4, comparison of performance of our method to that of other modeling approaches in terms of perplexity is presented. Section 5.5 concludes the paper and outlines future work.

5.2 Related Works

The seminal work on Latent Dirichlet Allocation (LDA) [41] provides basis for numerous extensions and generalizations in the field topic modeling. LDA considers document collections as bag-of-words assemblies that are generated by stochastic processes. To generate a document, a random process first selects a topic from a distribution over

topics and then generates a word by sampling the associated topic-word distribution. Both the topic and the word distributions are governed by hidden (or latent) parameters.

The LDA framework is designed to operate on a fixed set of documents and cannot be applied to stream data directly as converting an unbounded number of documents to a finite collection is not possible. To overcome this challenge, many approaches limit the training scope by aggregating messages based on attributes such as authorship or hash tag annotations and training models based on these aggregates [49],[50], [51].

An interesting recent work by Want et al. introduced an efficient topic modeling technique called TM-LDA for stream data. This approach is based on the notion that if document topic model is known at time t, at time $t + 1$ a new topic model can be predicted and an error can be computed by comparing the old and the new topic models. This error computation reduces the challenge of estimating topic models for new documents to a least-squares problem, which can be solved efficiently. Focusing on the popular Twitter micro-blog data, TM-LDA selects a set of individual authors and trains a separate model for each of the authors. To accomplish this, TM-LDA monitors Twitter for an extended period of time (a weeks worth of data was collected in the original work) and then trains a model to be able to predict new messages.

The idea of using authorship to improve topic modeling quality is not unique to TM-LDA. A recent work by Xu et al. modified the well-known Author-Topic [52] model for Twitter data [50]. Xu et al. extended the insight of the Author-Topic model by taking advantage of additional features available in Twitter such as links, tags, etc.

Another way to approach topic modeling in streams is to apply LDA machinery to snapshots or buffers of documents of fixed size. Online Variational Inference for LDA [53] is one such technique. The algorithm assembles mini-batches of documents at periodic intervals and uses Expectation Maximization (EM) algorithm to infer distribution parameters by holding topic proportions fixed in the E-step and then recomputing topic proportions as if the entire corpus consisted of document minibatches repeated numerous times. Topic parameters are then adjusted using the weighted average of previous values of each topic proportion.

Another approach termed On-line LDA [51] considers the data stream as a sequence of time-sliced batches of documents. The approach processes each time-slice batch using the classical LDA sampling techniques, with the variation being that the corresponding collapsed Gibbs sampler initialization is augmented with the inclusion of topic-word counters from histories of pervious time-slice batches. The histories are maintained using a fixed-length sliding window and the contribution of each history to the current

slice initialization is predicated upon a set of weights associated with each element in the sliding window.

In another work, Yao et al in [48] considered topic discovery in streaming documents and proposed the SparseLDA model. Noticing that the efficiency of sampling-based inference depends on how fast the sampling distribution can be evaluated for each token, their work enhanced the inference procedure in a way as to allow parts of computations used in sampling to be pre-computed, thus improving performance. Further, the sampling procedure proposed by Yao et al. restricted training to a fixed collection of training documents and then, for each test document, sampled topics using counts from the training data and test document only, ignoring the rest of the stream.

The explosion of micro-blog popularity has attracted much attention from outside of the topic modeling community. One particularly interesting application is the field of first story detection. Conceptually, first story detection is concerned with locating emergent clusters of similar stream messages, which are said to be indicative of particularly interesting and currently relevant stories. First story detection approaches require the ability to discover clusters of similar documents in near real-time fashion, which is difficult to accomplished using classic clustering tools since the computational complexity of commonly used clustering algorithms (hierarchical, partitioning, etc.) is quite high. Therefore, recent works on first story detection have seized upon the concept of Locality Sensitive Hashing (LSH) [49], which is an approach for identifying a datum neighborhood in constant time [54]. In [54], Petrovic et al use a combination of LSH and inverse index searching to show that clusters of similar documents may be identified in constant time with exceptional accuracy and low variability.

5.3 Hash-Based Stream LDA

As noted in the preceding survey of related works, many approaches to topic modeling in streams have been developed in recent years. A number of these approaches [47, 53] attempted to enhance quality by preserving various aspects of topic inference calculations and predicating topic learning upon past knowledge. Unfortunately, none of these techniques were successful in retaining the knowledge of stream documents relying instead on storing global structures such as topic-word multinomials. Hurdles for retaining document knowledge are two-pronged 1) the number of documents in streams is unbounded making storage of individual document information not feasible, and 2) since previous documents do not get replayed in streams, retaining records of their presence directly may be meaningless for topic modeling.

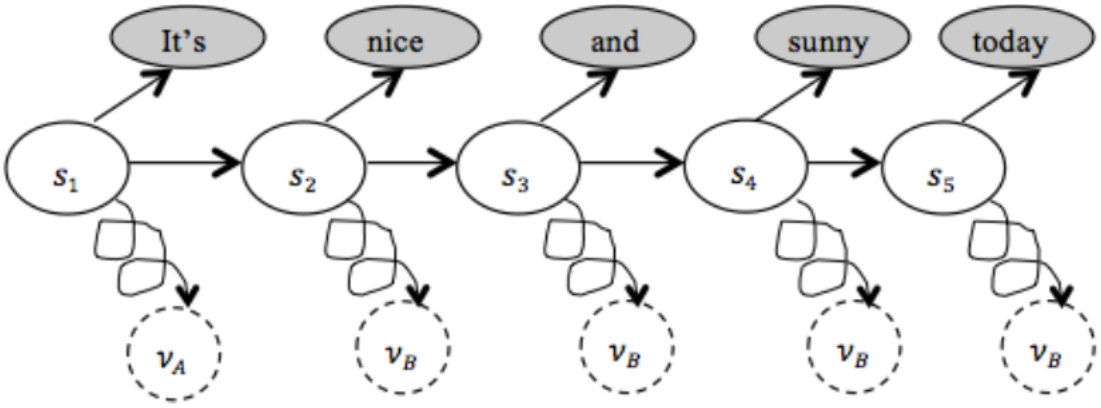


FIGURE 5.1: Visualization of the HS-LDA generative process. Ovals $_1, \ldots, s_5$ represent process states, shaded ovals represent word generation and dashed circles represent emissions of neutrinos ν of types A and B. Dashed circles surrounding neutrinos labels aim to emphasize the notion that neutrinos are assumed to be present but difficult to detect.

Therefore, this section introduces the new Hash-Based Stream LDA (HS-LDA) model, which provides a mechanism for retaining document knowledge for stream modeling in a scalable and meaningful way. HS-LDA is a generative probabilistic model that describes a process for generating a document collection. Like LDA, in HS-LDA each document is viewed as a mixture of underlying topics and each word is generated by drawing from a topic-word distribution. HS-LDA departs from LDA by imagining that, in addition to words, the generative process also emits certain auxiliary objects that are not directly observable in data. Since the auxiliary objects postulated by the HS-LDA model are not observable, we introduce the notion of HS-LDA neutrinos (or pseudo-neutrinos for short), as the analogy with the real-world ethereal particle seems appropriate.

Following the analogy with the physical particles [55], we consider the HS-LDA pseudo-neutrinos as belong to a fixed set of possible types (or flavors). The physics analogy is abandoned at this point, however, as HS-LDA makes no further claims as to the properties or nature of each flavor. The generative process is graphically outlined in Figure 5.2

In Figure 5.3, the generative process is outlined. There, words are generated in a way common to many LDA-type models by drawing from a distribution over words. Unlike other approaches, however, a pseudo-neutrino is also emitted by a draw from a multinomial distribution parameterized by a vector of topic-specific neutrino type proportions.

It is important to note that if a user were to restrict the set of possible neutrino types to just a single type (say root), HS-LDA would become equivalent to LDA as all draws of type label assignments would be the same making the generative branch from to redundant. Therefore, HS-LDA is a generalization of Latent Dirichlet Allocations, which

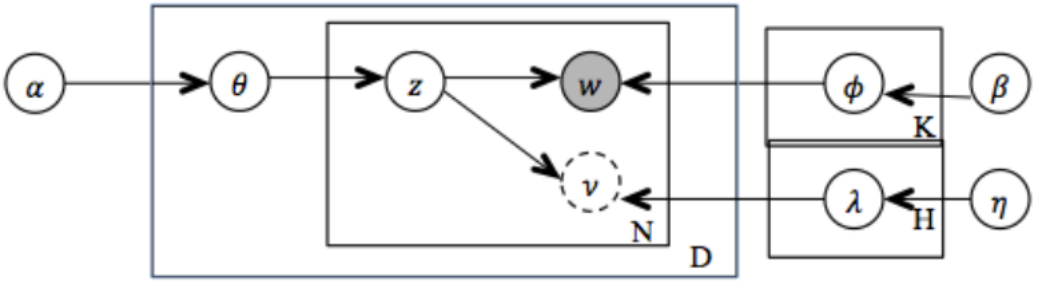


FIGURE 5.2: Graphical model representation of HS-LDA. N is the number of words in a document, D is the number of documents, K is the number of topics and H is the number of pseudo-neutrino types. α, η and β are Dirichlet prior vectors that are assumed to be symmetrical in this paper. represents the vector multinomial over topics, ϕ is the multinomial over words, z is the topic draw, w stands for a word realization and ν is the emitted pseudo-neutrino. The clear circles represent hidden entities, shaded circles represent directly observable entities and the dashed circles stand for indirectly detectable ones.

1. For each topic $k \in \{1, \ldots, K\}$:
   a. Generate $\phi_k = \{\phi_{k,1}, \ldots, \phi_{k,V}\}^T \sim Dir(\cdot|\beta)$
   b. Generate $\lambda_k = \{\lambda_{k,1}, \ldots, \lambda_{k,H}\}^T \sim Dir(\cdot|\eta)$
2. For each document d:
   a. Generate $\theta^d \sim Dir(\cdot|\alpha)$
   b. For each $i \in \{1, \cdots, N_d\}$
      a. Generate $z_i \in \{1, \cdots, K\} \sim Mult(\cdot|\theta^d)$
      b. Generate $w_i \in \{1, \cdots, V\} \sim Mult(\cdot|\phi_{z_i})$
      c. Generate $v_i \in \{1, \cdots, H\} \sim Mult(\cdot|\lambda_{z_i})$

FIGURE 5.3: Generative process for HS-LDA: ϕ_k is a vector consisting of parameters for the multinomial distribution over words corresponding to kth topic, λ_k is a vector consisting of parameters for the multinomial distribution over neutrino types corresponding to kth topic, α is the Dirichlet document topic prior vector, β word prior vector, η is the neutrino type prior vector and N_d is the number or words in document d and K is the number of topics.

is important to note since the general nature of HS-LDA suggests that its insight can be applied to other models that extend LDA, of which there are many. Later sections will take advantage of this fact and show the experimental results of application of HS-LDA to other successful models.

5.3.1 Gibbs Sampling with HS-LDA

The generative probabilistic HS-LDA model describes the process of document collection creation. The hidden model parameters θ, ϕ and λ may be estimated using a Monte Carlo

procedure, which is relatively easy to implement, does not require a lot of memory and produces output that is competitive with that of other more complicated and slower algorithms [47],[56]. The rest of the section describes the derivation of an efficient sampling algorithm used to infer models parameters with HS-LDA.

We start by framing the problem of topic discovery in terms of collections of D documents containing K topics expressed over W words and H pseudo-neutrino types. The task of learning topic models is to discover the makeup of θ, ϕ and λ, which can be estimated by evaluating the probability of a topic having observed both a word and a pseudo-neutrino. The posterior distribution is formally stated as Equation 5.1

$$P(z|w,\nu) = \frac{P(w,z,\nu)}{\sum_z P(w,z,\nu)} \tag{5.1}$$

The joint distribution $P(w,\nu,z)$ can be computed by considering that Dirichlet priors α, $\beta$and η in the HS-LDA model are conjugate to η, ϕ and λ respectively. Since $P(w,\nu,z) = P(w|\nu,z)P(\nu|z)P(z)$ by the chain rule and since w and $\nu$are conditionally independent in our model, $P(w|\nu,z) = P(w|z)$,which simplifies the joint distribution in Equation 5.2

$$P(w,\nu,z) = P(w|z)P(\nu|z)P(z) \tag{5.2}$$

Observing that ϕ, λ, and θ only appear in first, second and third terms respectively, each term may be evaluated separately. Integrating out ϕ, λ, and θ in each term gives Equations 5.3-5.5

$$P(w|z) = \left(\frac{\Gamma(W\beta)^K}{\Gamma(\beta)^W}\right)\prod_{j=1}^{K}\left(\frac{\prod_w \Gamma(n_j^w+\beta)}{\Gamma(n_j^{\cdot}+W\beta)}\right) \tag{5.3}$$

$$P(\nu|z) = \left(\frac{\Gamma(H\eta)^K}{\Gamma(\eta)^H}\right)\prod_{j=1}^{K}\left(\frac{\prod_\nu \Gamma(n_j^\nu+\eta)}{\Gamma(n_j^{\cdot}+H\eta)}\right) \tag{5.4}$$

$$P(z) = \left(\frac{\Gamma(K\alpha)^D}{\Gamma(\alpha)^K}\right)\prod_{d=1}^{D}\left(\frac{\prod_j \Gamma(n_j^d+\alpha)}{\Gamma(n_{\cdot}^d+K\alpha)}\right) \tag{5.5}$$

where n_j^w is the number of times word has been assigned to topic j, n_j^d is the number of time a word from document d has been assigned to topic j, n_j^ν is the number of times a neutrino of type ν has been assigned to topic j, $n_j^{\cdot}$ and $n_{\cdot}^d$ are the total numbers of assignments in topic j and document d respectively. $\Gamma(\cdot)$ is the standard gamma function.

Since computing the exact distributions in Equations 5.3-5.5 is intractable [56]], we follow the pattern in other topic modeling approaches and estimate θ, ϕ and λ by relying on the Gibbs sampling procedure. The Gibbs procedure operates by iteratively sampling all variables from their distributions conditioned on their current values and data and updating variables for each new state. The full conditional distribution $P(z_i = j|z_{-i}, w, \nu)$ that is necessary for the Gibbs sampling algorithm is obtained by probabilistic argument [56] as well as by observing that first terms in each of the Equations 5.3-5.5 are constant and values of denominators and numerators of second terms are proportional to the arguments of their gamma functions. Therefore, the sampling equation is as follows:

$$P(z_i = j|z_{-i}, w, \nu) \propto \frac{n^{w_i}_{-i,j} + \beta}{n^{\cdot}_{-i,j} + W\beta} \frac{n^{d}_{-i,j} + \alpha}{n^{\cdot}_{-i,j} + K\alpha} \frac{n^{\nu_i}_{-i,j} + \eta}{n^{\cdot}_{-i,j} + H\eta} \tag{5.6}$$

where, $n^{\nu_i}_{-i,j}$ is the count of times neutrino ν_i has been assigned to topic j excluding current assignment and $n^{\cdot}_{-i,j}$ is the total number of topics j assignments in any document excluding current assignment. Reader may notice that denominators in the first and third product terms in Equation 5.6 have identical counters. That is because, in the HS-LDA model, the number of words is always exactly the same as the number of neutrino emissions by process construction.

The Gibbs sampling algorithm can be implemented in an on-line fashion by first initializing topic assignments to a random state and then using Equation 5.6 to assign words to topics. The algorithm operates by reconsidering data for a number of iterations during which new states of topic assignments are found using Equation 5.6. The algorithm is fast as the only information necessary to estimate the new state is the word, topic and neutrino counters, which can be cached and updated efficiently [56].

5.3.2 "Neutrino" Detection

The sampling algorithm outlined in the previous section estimates parameter values by relying on two detectable quantities words and pseudo-neutrino emissions. To detect the latent auxiliary particles that cannot be observed directly in text, we assumed a Gaussian distribution of pseudo-neutrinos in documents, as this distribution was common to many phenomena [14]. With this assumption, we could refer to all pseudo-neutrinos in a given document in a meaningful way by identifying the most common (or mean) neutrino type. That is, for $H \in \mathbb{Z}^+$ possible pseudo-neutrino types, we assumed that there existed a mean pseudo-neutrino type $1 \leq c^d_\nu \leq H$ for each document d. With that, a rough

approximation vector of pseudo-neutrino assignments $h_d = \{h_{d,1}, \ldots, h_{d,H}\}$ could be constructed for each documentof size N_d such that $h_{d,i} = \begin{cases} N_d & \text{if } i \in c_\nu^d \\ 0 & \text{otherwise} \end{cases}$.

Constructing the vector h_d as described in the previous paragraph suggested that a meaningful approximation of document pseudo-neutrinos could be found by identifying a representative (mean) neutrino type for each document. To locate the representative flavor, we noticed that pseudo-neutrino types essentially constituted a kind of vocabulary akin to that of words. With that, considering topics from conceptual point of view, intuitively, documents on the same topic would be close to one another in terms of similarity of their content regardless of the vocabulary used to express the content (e.g. for any language, documents about the World Cup sporting event would contain text related to the even in that language). With that, since the number of pseudo-neutrino types was known, clustering documents into H clusters based on word similarity would approximate document-level (mean) neutrino types as cluster indices could be used as the neutrino type identifiers.

To implement this intuition in practice, we searched for a clustering strategy that would perform in a scalable way while at the same time ensuring that similar documents were likely to share a cluster. We realized that by restricting $H = 2^n$ for some positive integer n, it would be possible to make use of Locality Sensitive Hashing (LSH) [46].

LSH relies on existence of a set of hash functions H (referred to as a function family) for some d-dimensional coordinate space $\mathbb{R}^d$ where each hash function can be efficiently implemented with the help of Random Projections (RP) [57]. To use LSH, we start by defining a function space $f : \mathbb{R}^+ \to \{0, 1\}$ and constructing a function family $H = \{f_1, \ldots, f_{log_2(H)} | f_i \in f\}$. Each function ! H is associated with a random projection vector !random R! with components that are selected at random from a Gaussian distribution $N(0, 1)$. Each random projection is used to compute a dot-product between it and any point $p \in \mathbb{R}^+$ allowing the mapping function to be constructed in the following way:

$$h_{d,i} = \begin{cases} 1 & \text{if } p \cdot p_i^{random} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

Then, for any $p \in \mathbb{R}^+$, LSH hash value is constructed by invoking each of the functions in H on p and concatenating output bits as a bit string. Treating the bit string as a binary number, a mapping function assigns p to a number between one and H as follows:

$$map(p) = ||_{i=1}^{|H|} f_i(p) \tag{5.8}$$

Since the bit string generated by the above procedure is of finite size, the space of possible values is bound by $2^{|H|}$. Recalling that $H = 2^{|n|}$ and $|H| = log_2(H) = n$, function *map* can be used to map each point in $\mathbb{R}$ to a positive integer bound by H.

Further, since it is proven in [58] (proof omitted here) that $P(f_i(p) = f_i(q)) = 1 - \frac{angle(p,q)}{\pi}$ holds for any function $f_i \in H$ and all points $p, q \in \mathbb{R}^d$, the probability of LSH hash collision for two vectors increases with the decrease to the angle between them. Then, since the value of cosine of two vectors is directly related to the size of the angle

$$P(f_i(p)) = f_i(q) \propto cos(angle(p, q)) \tag{5.9}$$

where *angle* is the angle between the two vectors in radians[1].

Therefore, since LSH hashing allowed for fast clustering of vectors in a way that preserved document similarity, LSH was used to approximate the mean pseudo-neutrino type by treating LSH hash value as the type identifier. To make use of LSH hashing in topic modeling, we restricted the size of the set H to be a power of two and rewrote the sampling equation (Equation 5.6) in terms of LSH hash family F of size $log_2(H)$ as:

$$P(z_i = j|z_{-i}, w, z) \propto \frac{n_{-i,j}^{w_i} + \beta}{n_{-i,j}^{\cdot} + W\beta} \frac{n_{-i,j}^{d} + \alpha}{n_{-i,j}^{\cdot} + K\alpha} \frac{n_{-i,j}^{h_d^F} + \eta}{n_{-i,j}^{\cdot} + H\eta} \tag{5.10}$$

where h_d^F is the LSH hash value of document d, $n_{-i,j}^{h_d^F}$ is the number of words from documents with hash h_d^F assigned to topic j excluding current assignment, and $n_{-i,j}^{\cdot}$ is the total number of words in any document assigned to topic excluding current assignment. The sampling algorithm, then, proceeds as outlined in section 5.3.1 using Equation 5.10 to assign words to topics.

5.4 Evaluation

In order to validate the utility of our model, the approach was tested on two distinct data sets. Our first data set consisted of 1,000,000 English language messages collected from Twitter micro-blog site using its public sampling API over a period of one week. The second data set was comprised of 300,000 English language chatroom messages

[1] Unusual angle operator used to avoid confusion with topic modeling notation

collected by connecting to the public *irc.freenode.net* public chat server and monitoring chat rooms with more than 150 chatters for the same one week period. Filtering of non-English texts was accomplished with the help of the open source language-detection[2] library.

The language models produced by our approach were compared to those learned by On-line LDA and SparseLDA as these models were designed to operate efficiently on stream data. In addition, to provide a common baseline, topic models learned by HS-LDA were compared to those discovered by the classic LDA algorithm. We did not evaluate our approach against TM-LDA as it required partitioning by author as well as a significant and static training sample to be collected prior to producing any output at all. These constraining requirements made TM-LDA unfit for continuous topic modeling application, which was the motivation of this work.

To compare language models, evaluation was performed using the perplexity measure over held-out subset of data. The perplexity measure was used as described in 4.5.3 in the previous chapter.

5.4.1 Parameter Selection

As pointed out in earlier works [52, 59], Locality Sensitive Hashing is highly sensitive to choices of the hash family size. This choice governs the scatter within each hash bucket as chance of collision decreases with the increase of hash family size. Therefore, hash family size selection was approached from the point of view of estimating a reasonable number of buckets for the number of messages expected.

Considering the Twitter micro-blog service as being one of the most vibrant and popular social forums today, we experimented with the numbers of English language messages that could be downloaded over a given period. Recalling the industry-oriented motivation for this work and selecting one working week as the target period (time-frame common to the industry environment) the number of messages that could be gathered from Twitters sampling service was empirically estimated to number in some millions. Realizing that if the number of hash family function was chosen to be high (ex.: $2^{20} = 1,048,576$) the algorithm could potentially map every message into an individual bucket, negating the entire insight of HS-LDA. With that, the reasonable number of hash functions for our experiments was chosen to be 17 ($2^{17} = 131,072$) as this number would allow for variability within each cluster while at the same time providing reasonable specificity.

[2] https://code.google.com/p/language-detection/

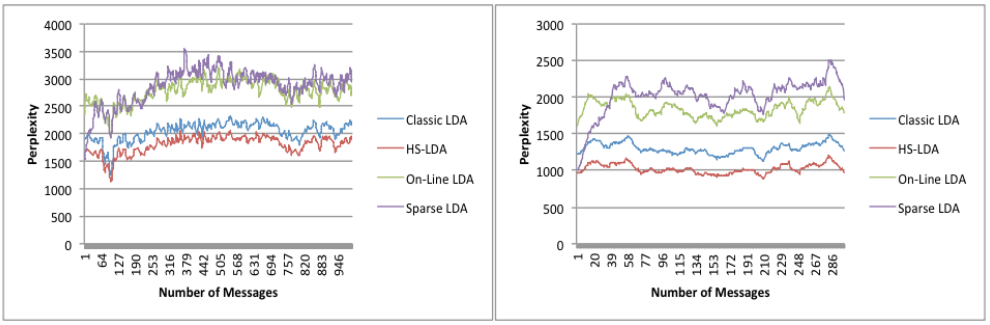


FIGURE 5.4: Smoothed perplexity results for Twitter (left) and IRC (right) dataset

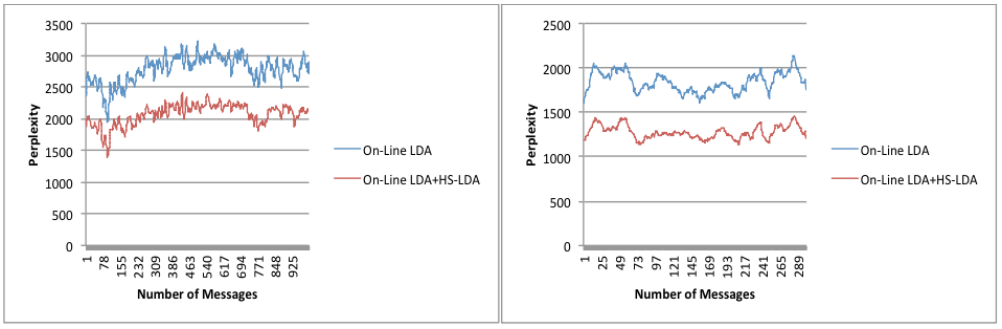


FIGURE 5.5: Pairwise comparison of On-Line LDA and On-Line LDA augmented with HS-LDA for Twitter (left) and IRC (right) test sets

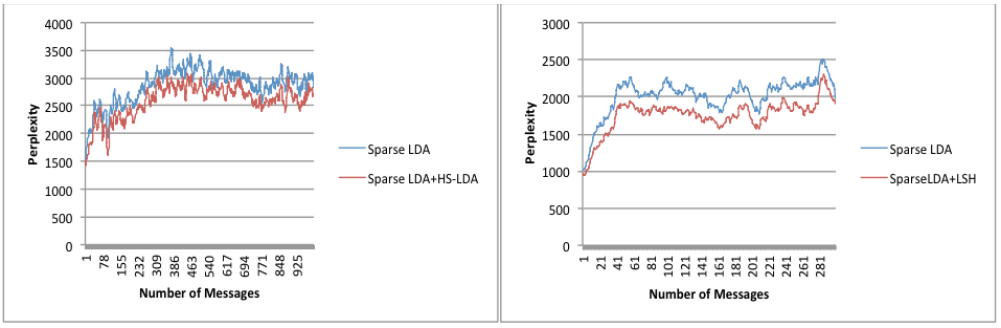


FIGURE 5.6: Pairwise comparison of Sparse LDA and Sparse LDA augmented with HS-LDA for Twitter (left) and IRC (right) test sets.

5.4.2 Experimental Setup and Results

Having thus chosen the hash family size, HS-LDA was evaluated against LDA, On-line LDA and Sparse LDA using the two test datasets. For all models, the number of topics was chosen to be 100 and experimented with various hyperparameter settings. Results reported here were for hyperparameter values of $\alpha = 0.05$, $\beta = 0.05$ and $\eta = 1$ as these values produced best results for all models.

Figure 5.4 shows perplexity results for the two test datasets. In order to provide a readable graphic, the Simple Moving Average (SMA) smoothing technique was applied to raw results, setting the moving average window set to 10,000.

To summarize results in numerical way, average perplexities are reported for all tested models in Table 5.1. The purpose of this report is to identify the model with the highest predictive prowess as well as to quantify amount of improvement in terms of percentages.

Model	**Average Perplexity (Twitter)**	**Average Perplexity (IRC)**
LDA	2044.42	1300.92
On-Line LDA	2773.99	1835.74
Sparse LDA	2860.27	1998.53
HS-LDA	**1803.67**	**1023.12**

TABLE 5.1: Average perplexity results for Twitter and IRC datasets

In Table 5.1, HS-LDA outperformed other models by at least approximately 12% for the Twitter dataset and 21% for the IRC chatroom data. Significantly better predictive power of resulting topic models learned from the chatroom discourse may be explained by noting that chatrooms are often oriented towards particular themes, thus introducing loose structuring to social discourse. Such structuring does not exist in Twitter where the discourse is entirely unstructured, making the job of theme discovery more difficult.

5.5 Conclusions

To improve the quality of topic models learned from social media streams, we introduced the new HS-LDA model for topic modeling, which was a generalization of the well-known LDA topic discovery technique. We experimented on large data sets collected from popular social media services and showed that our model outperformed other state-of-the-art stream topic modeling techniques in all cases. Further, we enhanced other topic modeling approaches with the insight of HS-LDA and showed that applying core notions of HS-LDA to other techniques improves their performance in terms of predictive power of resulting topic models.

While our results showed improvement in all cases where HS-LDA insight was used, combining HS-LDA with other models aimed at preserving global context did not immediately result in substantial performance gains. It seems, however, that such a combination has merit and we will continue this investigation in the future work.

Further, while this work was instrumental in moving towards the goal of constructing an industry-grade stream topic monitoring system, one of the major hurdles for constructing such a system with HS-LDA was the necessity to specify the number of topics. In our future work, we plan to investigate topic modeling approaches based on the popular Chinese Restaurant Process paradigm and will attempt to apply the insight of HS-LDA to dynamically discovered topic allocations.

Chapter 6

Microblog-hLDA: Semi-Parametric Hierarchical Topic Modeling in Microblogs

Topic modeling approaches, such as Latent Dirichlet Allocation (LDA) and Hierarchical LDA (hLDA) have been used extensively to discover topics in various corpora. Unfortunately, these popular techniques do not perform well when applied to collections of social media posts. We argue that the poor performance is the result of data sparsity in short and noisy microblog messages and introduce the new Microblog-hLDA generative model that, unlike fully non-parametric approaches, such as the hLDA, generates text in a way that allows for the inverse power law (Zipf's Law) assumption to be made about the resulting corpus. We show that this assumption is helpful in hierarchical topic modeling by comparing topic models learned with our approach to those discovered by Hierarchical LDA, Tree-Structure Stick Breaking (TSSB) and Recursive CRP(rCRP). We apply Microblog-hLDA to two Twitter collections and a corpus of IRC chatroom messages and show that our model outperforms others in terms of log-likelihood of held-out data. Further, we introduce a new metric to quantify specificity of words in topic hierarchies. The new metric is used to augment the topic specialization measurement when comparing topic hierarchies discovered with Microblog-hLDA against those produced by TSSB, hLDA and rCRP. The results show that topic hierarchies discovered by Microblog-hLDA smoothly increase in specialization towards the leafs – pattern that is not observed in TSSB, hLDA and rCRP.

While, admittedly, the resulting algorithm in this chapter is as an apparently simple modification to hLDA, the theoretical argument for this simple modification was an important step in my work towards development of more sophisticated extensions presented

in subsequent chapters.

6.1 Introduction

We study hierarchical topic modeling in microblogs. Microblogs are popular social media outlets where users publish short, free-form text messages targeted at a certain group of friends or aimed at larger audiences. Microblogging social media systems, such as Twitter and Facebook, have become very popular in recent years. These systems are frequented by millions of users that author volumes of original content.

We attempt to discover meaningful constructs in this content through hierarchical topic modeling. Topic modeling is a name for statistical techniques that automatically discover topics (defined as probability distributions over words) in data. Hierarchical topic modeling takes it a step further and tries to expose interesting relationships among topics by organizing them as hierarchies.

It has been reported by many researchers that current topic modeling approaches perform poorly when applied to collections of social media messages [60] [61] [62][63][64][65]. One possible reason for the poor performance may be that many common topic modeling techniques are non-parametric – designed to learn model parameters entirely from data. While non-parametric approaches have been shown to work well for larger documents, such as newspaper articles or scientific papers [66], short and noisy microblog messages may not have enough content for practical non-parametric inference.

Therefore, in this paper, we propose a novel hierarchical generative model called Microblog-hLDA, which generates text in a way that allows the learning procedure to take advantage of a parametric assumption when discovering topic structures. We evaluate the new model against related approaches that include Hierarchical LDA[66], Tree-Structure Stick Breaking[67] and Recursive Chinese Restaurant Process [68]. We compare these algorithms in terms of heldout log-likelihood and topic specialization using three large social media data sets and show that Microblog-hLDA outperforms others by a significant margin. Further, we propose a new metric called the *expected topic rank*, which measures word specificity across hierarchy levels. Evaluation using the proposed metric shows that topics specialization increases smoothly towards leafs in Microblog-hLDA – pattern that is not observed for other approaches.

The paper is organized as follows. Section 7.2 discusses the current state of research in the area of topic modeling in general and microblog topic modeling in particular. Section 7.3 offers an analysis of topic modeling challenges in social stream data and describes the new Microblog-hLDA model designed to overcome these challenges. In

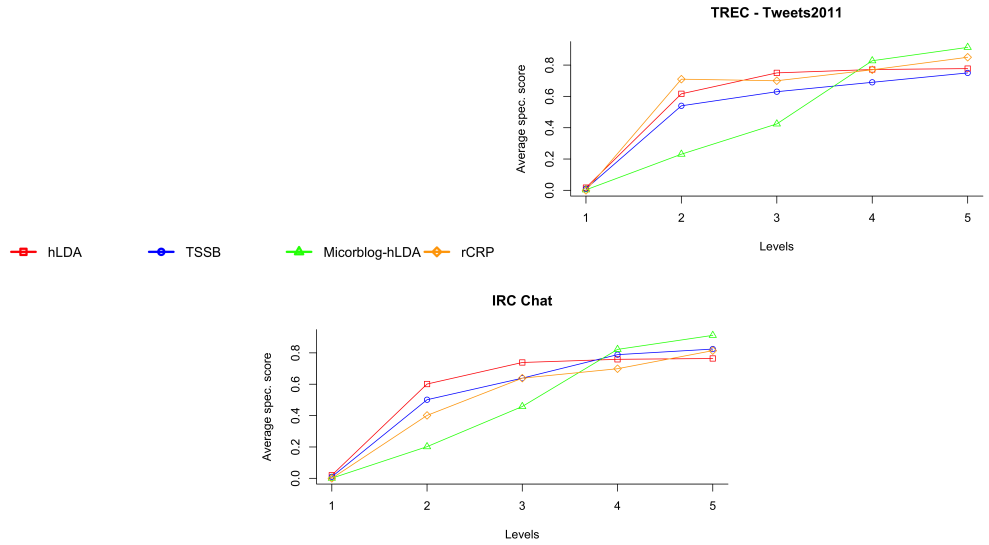


FIGURE 6.1: Topic specialization scores for Microblog-hLDA, hLDA, TSSB and rCRP showing the ability of Microblog-hLDA to find progressively more specialization topics proportional to the distance from the root

Section 7.4, we discuss data sets and experiments that were used to evaluate how well the new approach performs as compared to others. Section 7.5 concludes the paper and outlines future work.

6.2 Related Works

Topic modeling in text aims to discover hidden relationships between words. That is, for example, if words *'pizza'* and *'pasta'* never occur together in any document in some corpus, a topic modeling technique is expected to co-locate these words near each other in terms of their probabilities in a topic, thus discovering a non-obvious relationship. If these words were *representative* of a topic (i.e.: highly probable as related to other words), a human evaluator could quickly grasp its theme and label it, perhaps, as *'Italian Food'*.

The Latent Dirichlet Allocation (LDA) [69] framework has become a popular choice for topic modeling in recent years. This popularity has often been attributed to the flexibility and modularity of LDA, which easily lends itself to extensions and generalizations that accommodate many types of relationships in data [70].

LDA is a generative probabilistic model that makes the "Bag-of-Words" assumption and represents documents as probability distributions over K topics. These topics are, in turn, viewed as probability distributions over W words. In LDA, for a corpus of D

documents, the probability of a word w in a document is given by

$$p(w|\theta, \beta) = \sum_{z} p(w|z, \beta)p(z|\theta) \tag{6.1}$$

where θ is a document-specific K-dimensional topic mixture, β is a $K \times W$ matrix such that $\beta_{ij} = p(w^i = 1|z^i = 1)$ and z is a topic. [69]

In LDA, words are generated by randomly selecting a distribution over topics $\theta_{t|d}$ for each document $d \in D$. Then, for each i^{th} word in d, a topic assignment, z_{id}, is drawn from $\theta_{t|d}$ and the word, x_{id}, is drawn from the corresponding topic, $\phi_{w|z_{id}}$. The generative LDA model is given as

$$\begin{aligned} \theta_{t|d} &\sim Dir(\alpha) \qquad \phi_{w|t} \sim Dir(\beta) \\ z_{id} &\sim Mult(\theta_{t|d}) \;\; x_{id} \sim Mult(\phi_{w|t}) \end{aligned} \tag{6.2}$$

where α and β are Dirichlet prior vectors [69].

While LDA has enjoyed much popularity serving as basis for numerous extensions and generalizations, one of its major limitations is that users must select the number of topics K before the approach can be used. This requirement makes the approach quite rigid, as it cannot accommodate influx of new data [66]. To make topic modeling more flexible, LDA machinery was modified in [66] to use the Chinese Restaurant Process (CRP) [71]. CRP relaxes the fixed K constraint of LDA by assuming an infinite number of topics and postulating that words are generated from topics according to the following distribution:

$$\begin{aligned} p(existing\ topic\ i|previous\ words) &= \frac{m_i}{\lambda + m - 1} \\ p(new\ topic|previous\ words) &= \frac{\lambda}{\lambda + m - 1} \end{aligned} \tag{6.3}$$

where m_i is the number of words assigned to topic i, λ is a parameter and m is the total number of words seen so far. The formulation in Equation 7.1 removes the need to know K *apriori*, as it assigns a non-zero probability to choosing a new topic. This allows the number of discovered topics to grow as the new data arrives.

While the CRP approach automatically discovers the set of topics, ability to grasp meaningful insights is greatly enhanced if those topics are presented as hierarchical structures [1]. Organizing topics into hierarchies may be particularly important for microblog data, as this data is very diverse and may contain many disjoint themes, which could be difficult to evaluate as a simple collection.

The Hierarchical LDA (hLDA) model proposed by Blei et. al. [66] exposed a way to learn topic hierarchies from data. The hLDA generative probabilistic model assumes that words in a document are generated from an infinitely branched tree of height L

according to a mixture model that is random and document-specific. In hLDA, each node of the tree is associated with a single topic.

To learn topic and tree structure from data, hLDA inference starts by choosing an L-level path c_d for each document d according to:

$$p(\boldsymbol{c}_d|\boldsymbol{w}, \boldsymbol{c}_{-d}, \boldsymbol{z}) \propto p(\boldsymbol{w}_d|\boldsymbol{c}, \boldsymbol{w}_{-d}, \boldsymbol{z})p(\boldsymbol{c}_d|\boldsymbol{c}_{-d}) \tag{6.4}$$

where $\boldsymbol{w}_{-d}$ and $\boldsymbol{c}_{-d}$ are words and paths of documents other than d; $\boldsymbol{w}$ and $\boldsymbol{c}$ are words and paths of all documents, respectively; $\boldsymbol{z}$ is the topic assignments. Since hLDA associates each tree node with a single topic, a sampling equation may be used to infer document-specific distribution parameters over L topics. [66]

The seminal hLDA model provided inspiration to many works in recent years. One such work is the Tree-Structured Stick Breaking (TSSB) process introduced by Adams et al. [67]. Unlike hLDA, TSSB imagines that each document is generated from a single node of an infinite tree and each node is associated with a distribution over K topics.

Another recent work termed Recursive CRP (rCRP) [68] learns topic hierarchies by using a recursive approach. Unlike hLDA and TSSB, rCRP models documents as originating from any branch of the latent hierarchy of topics. rCRP applies a recursive approach where each word is generated by a recursive dissent through an infinitely deep and infinitely branched hierarchy.

With the rise of microblog popularity, many researchers have focused their efforts on improving topic modeling for social media texts. Some works have attempted to improve results by aggregating messages based on similarity attributes. For instance, Xu et al. [72] modified the well-known Author-Topic [73] model to take advantage of additional attributes available in Twitter, such as links, tags and other features. In another work, Mehrotra et al. [60] proposed several pooling schemes that aggregated Twitter messages from the same user (or based on other attributes) into synthetic documents and used these documents to train LDA. In their experiments, Mehrota et al. showed that such aggregates significantly improved interpretability of LDA models.

Other works proposed changing the LDA machinery itself to accommodate the nature of microblog data. For instance, Zhao et al. modified the classical LDA generative model to use user-specific topic proportions rather than document-specific ones. [61]

While the previous works do a lot to improve topic modeling quality, they lack in several important areas. Some of the previous efforts base their improvements on leveraging non-textual context, such as authorship, social tagging, time, etc. Such approaches are limiting in that they require careful selection of attributes[60], as well as an excellent

understanding of the target social media system semantics. In addition, none of the previous approaches offer a solution for extracting quality hierarchical views of topics from microblogs.

Further, numerous researchers have reported that topic modeling techniques based on LDA under-perform in short and noisy corpora, such as the microblogs [60] [61] [62][63][64][65]. Experiments suggest that topics learned from microblogs by LDA-style approaches are difficult to interpret for human reviewers [60].

In this paper, we aim to solve the aforementioned challenges by discovering a semi-parametric approach that aims to improve the quality of hierarchical topic mining in microblogs. Unlike previous works that leverage non-textual context, we strive to keep our approach general enough to be applied to any microblog data without the need to select or group messages based on carefully chosen contextual attributes.

6.3 Microblog-hLDA Model

The fully non-parametric nature of hLDA allows the data to "speak for itself", which makes it a popular choice for many applications. Unfortunately, the "speach" may be muffled in microblog context. hLDA uses sampling during inference to find a hidden tree node for each observed document and to guess which of the node's parents generated which of the document's words. While millions of new microblog messages make non-parametric estimation of path probabilities feasible, approximating document-specific topic proportions from the few words in a Facebook status update may be a challenge.

In more concrete terms, we found that messages in the Tweets2011 data set used in this study contained, on average, 12.3 words per message. When estimating topic proportions for a 5-level hierarchy with hLDA, a non-parametric regression procedure would have to rely on approximately 2.5 data points per each of the 5 possible topics, which is, perhaps, too sparse for meaningful estimates.

In practice, data sparsity is often tackled by making assumptions about the data. If these assumptions are well-grounded, relatively few data points are often sufficient to achieve acceptable results. With that, we recall that word frequencies in natural language texts have been shown to follow the inverse power law distribution, known as the Zipf's Law [74]. In other words, when word ranks are plotted against their frequencies at *log* scale, a near linear relationship is observed. We, therefore, imagine that the Zipfian distribution that is expected to be observed in microblog texts is the result of a generative process that outputs corpra in a way that suggests the inverse power law property. The rest of the section presents the new model, called Microblog-hLDA, which generates inverse

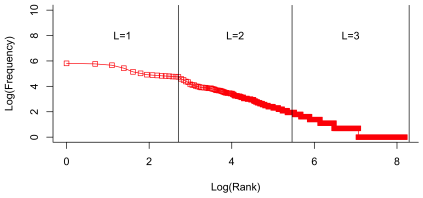


FIGURE 6.2: Example of partitioning 3604 distinct words from a sample corpus into level buckets with Equation 7.4. Vertical lines indicate level bucket boundaries. L stands for level indicator.

power law distributed text corpora. Once the new model is presented, we show how the Zipf's Law assumption may be used to ascertain topic assignments for each words logically, rather than through non-parametric regression. We then test our approach against other hierarchical topic modeling algorithms and show that Microblog-hLDA outperforms comparable techniques in terms of the quality of the resulting hierarchical topic models.

6.3.1 Generative Process

The process, which is outlined in Figure 6.3, begins by generating an inverse power law distributed collection of random strings. For that, we recall the well-known work by Li [**?**], which shows that, if a string is generated by randomly drawing characters from a finite alphabet, which contains a separator character (e.g.: blank space), the resulting collection of tokens (when the string is tokenized by the separator) can be proven to obey the inverse power law distribution. The proof, which we do not reproduce here, relies on the observation that short string of form "_abc_" are probabilistically more likely than the longer ones (e.g.: "_zyxwvutsrqp_").

Therefore, the Microblog-hLDA process starts by reproducing Li's random string generation algorithm in step 4 of the generative process. The resulting string is tokenized by the separator and the frequency table is constructed from token counts, in step 6. As proven in [**?**], random "words" in the frequency table must follow the inverse power law distribution when ranked according to their frequencies.

The algorithm then assumes the existence of an L-level hierarchy that emits the tokens assembled in the aforementioned frequency table in the following way. First, the algorithm partitions the *log*-scale plot of the tokens into L partitions with Equation 7.4 (see Figure 6.2 for a visualization), which is defined as

$$Level_w = \lfloor log_{\sqrt[L]{|V|+1}}(rank(w)) \rfloor + 1 \tag{6.5}$$

where V is the set of unique tokens after the random string tokenization and $rank(w)$ is the rank of token w in the frequency table. That is, for example, when considering a 3-level hierarchy ($L = 3$) and a 1000 term vocabulary ($|V| = 1000$), Equation 7.4 will associate the 10 most frequent terms with the root level, next 90 with the intermediate level, and 900 least frequent ones with the leaf nodes.

The process, then, constructs a set of L masks (one for each level) in step 4(g)e. The masks are constructed in such a way that, when the l^{th} mask is applied to a word proportions vector, all elements of that vector that do not correspond to words in the l^{th} partition are zeroed out. That is, from the example above, when 1^{st} level mask is applied to any topic vector, the resulting vector would contain no more than 10 non-zero entries, 2^{nd} level mask would result in at most 90 positive values and 3^{rd} level mask would limit word proportions to the maximum of 900 non-trivial components.

Once words are allocated to levels, the generative process activates the corpus generation logic in step 10. As in hLDA, Microblog-hLDA determines a path through the latent hierarchy with Equation 7.1 for each document. Then, for each word, the process randomly selects the l^{th} node on the path by a draw from a uniform distribution over L possibilities in step 10(2)a. Once a node is selected, the l^{th} mask is applied to the corresponding word proportions vector in step 10(2)c and a word is sampled from a distribution parameterized by the normalized masked vector in step 10(2)d.

We now argue that the generative process described above generates corpora that may be expected to follow the Zipf's Law of word frequencies. We start by considering cases where the Microblog-hLDA model may produce a corpus that does not obey the inverse power law. We quickly discard the situation where all documents may be generated by a single path – since Equation 7.1 assigns a non-zero probability for selecting new branches – and focus instead on a case where all latent topics at each level of the hierarchy happen to be exactly identical[1] and have most of their mass concentrated at a single word or a small group of words. In this case, since topics are drawn uniformly in step 10(2)a, all words in the resulting corpus would appear with similar frequencies, thus violating the Zipf's formula.

While possible, the above eventuality is highly unlikely because of the following argument. Assuming conditional independence of both topics and words, the probability that masked topic vectors β'_i and β'_j are identical for some topics i and j at the hierarchy level l is $P(\beta'_i = \beta'_j) = \prod_{v=1}^{|V|} P(\beta'^v_i = \beta'^v_j)$. Letting W_l be the set of words available at level l, since vector elements that correspond to words in $V \backslash W_l$ are always zero by construction, $\prod_{v=1}^{|V \backslash W_l|} P(\beta'^v_i = \beta'^v_j) = 1$ and, therefore, $P(\beta'_i = \beta'_j) = \prod_{v=1}^{|W_l|} P(\beta'^v_i = \beta'^v_j)$.

[1] While we do not do so here to conserve space, the argument may easily be modified to discuss similar topics, rather than identical ones

1. Let A be the set of all possible characters
2. Let $s \in A$ be a separator character (such as a blank space)
3. Let $W_{corpus} =<>$ be a string
4. For each document
    1. For each word $n \in \{1, \ldots, N\}$:
        1. For each letter $c \in \{1, \ldots, C+1\}$:
            1. Draw a character $ch \in A$ from $Uniform(A)$
            2. Set $W_{corpus} = W_{corpus} \| < ch >$
5. Tokenize W_{corpus} by the s character and construct vocabulary set V from resulting tokens
6. Construct a corpus frequency table F_{corpus} by counting occurrences of each token in V in the string W_{corpus}
7. Construct a set of masks $\mathbf{M} = \{M_1, ..., M_L\}$ where $M_i \in \mathbf{M}$ is an $|V| \times |V|$ diagonal matrix such that each j^{th} element of the diagonal $M_i^{jj} = \begin{cases} 1 & \text{if } Level_{V_j} = i \\ 0 & \text{otherwise} \end{cases}$ and $Level_{V_j}$ is as defined in Equation 7.4
8. Let c_1 be the root
9. Generate $\beta_{c_1} = (\beta_{c_1,1}, \ldots, \beta_{c_1,|V|})$ from $Dirichlet(\eta)$
10. For each document
    1. For each level $l \in \{2, \ldots, L\}$:
        a) Draw a child node form c_{l-1} using Equation 7.1. Set c_l to be that node
        b) Only once for node c_l, generate $\beta_{c_1} = (\beta_{c_1,1}, \ldots, \beta_{c_1,|V|})$ from $Dirichlet(\eta)$
    2. For each word $n \in \{1, \ldots, N\}$:
        a) Draw $z \in \{1, \ldots, L\}$ from $Uniform(\{1, \ldots, L\})$
        b) Let β_z be the word proportions vector associated with node c_z
        c) Let $\beta'_z = M_z \times \beta_z^T$
        d) Draw a word w from $Multinomial(\frac{\beta'_z}{|\beta'_z|})$

FIGURE 6.3: Microblog-hLDA generative algorithm

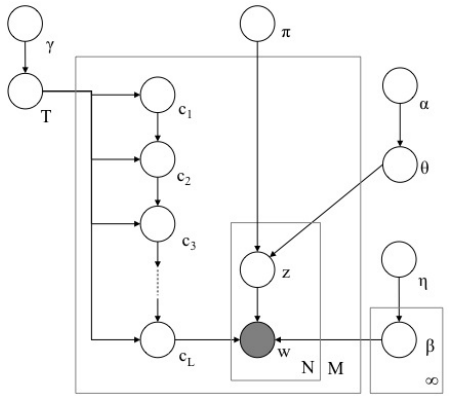


FIGURE 6.4: Microblog-hLDA Graphical Model

Since the number of words in W_l grows exponentially as l increases, the probability of topic-word proportions being exactly equal decreases exponentially. Therefore, the chance of drawing a set of identical topics at some level is quite small.

Then, since a word may only be generated from one and only one level in our model, the marginal probability of a word w at a given level l is $P(w|l) = \frac{\sum_{k=1}^{K_l} P(w|k)}{\sum_{w'=1}^{|W_l|} \sum_{k=1}^{K_l} P(w'|k)}$, where k is a topic and K_l is the number of topics at level l. Then, the probability of generating a word w is $P(w) = \sum_{l=1}^{L} P(l)P(w|l)$. Since Microblog-hLDA samples levels uniformly, $P(w) = \frac{P(w|Level_w)}{L}$.

Therefore, the probability of Microblog-hLDA generating a word is inversely proportional to the number of words available at its level, as governed by 7.4. Since, in our model, the number of words available at each level grows proportionally to the level index, words that are frequent in F_{corpus} are likely to be generated frequently by the process, whereas terms that are infrequent in F_{corpus} will be drawn infrequently. Since F_{corpus} is Zipfian by construction, the resulting corpus must exhibit the inverse power law property in the likely scenario where topics are reasonably well mixed.

6.3.2 Microblog-hLDA Inference

We now outline the inference procedure for Microblog-hLDA. The goal of posterior inference in topic modeling is to recover hidden model parameters from observed data. In our case, when presented with a corpus of microblog posts, we imagine that words in the corpus are tokens from the randomly generated string of characters W_{corpus} (see Figure 6.3) that just happened, by chance, to be words in a human language. With that assumption, inferring the structure of the hidden hierarchy and its topics is accomplished as follows.

As in hLDA, Microblog-hLDA samples an L-level path through a hierarchy for each document with the help of Equation 7.2. Then, since the frequency table of the observed microblog message collection is expected to resemble the hidden frequency table F_{corpus}, each word's level assignment is approximated directly with Equation 7.4 without sampling. By iteratively sampling paths for observed documents and updating appropriate counters, the inference procedure learns path proportions for each document non-parametrically from data, while choosing path elements for each word by making use of the Zipfian assumption, thus avoiding the need to estimate topic proportions from the few words in a microblog post with non-parametric regression.

6.4 Evaluation

In order to validate the utility of our model, the approach was tested on three distinct data sets. To facilitate repeatability of our results, we used the Tweets2011 Twitter Collection available publicly through the TREC project [75]. This data set consisted of 16 million Twitter messages sampled in early months of 2011. To verify that the usefulness of our approach was not limited to one particular data set, we collected 1,000,000 English language messages from the Twitter microblog site using its public sampling API over a period of one week.

Then, to ensure that our model was applicable to systems other than Twitter, we collected a third data set from an IRC chatroom system. While the IRC system was not technically a microblog, messages published on that system exhibited characteristics similar to posts found on microblog forums, such as Twitter or Facebook. We therefore collected 300,000 English language IRC chatroom messages by connecting to the public *irc.freenode.net* chat server and monitoring chat rooms with more than 150 chatters for the same one week period.

For each data set, message text was extracted and pruned of *hashtags* and user mentions (e.g.: "@user"). Punctuation and numeric symbols were removed as well. No stemming was performed as we found that the noisy and unedited nature of microblog texts caused stemming rules to heavily overstem (reduce words that should not be reduced) or understem (ignore words that should be stemmed). Word-length pruning (i.e.: removing words below a certain length) was not performed as we found that users often used short versions of words (e.g.: "u" instead of "you") and removing such words from already short microblog documents decreased topic readability in all tested topic modeling approaches. Further, no stopword removal was performed as it was expected that hierarchical topic modeling would automatically identify stopwords and highlight them by associating with the root node.

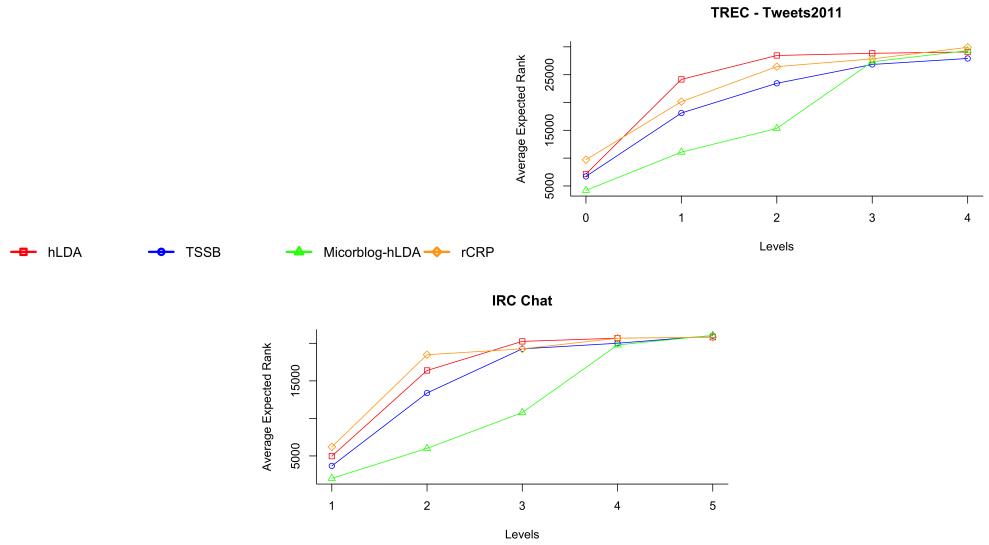


FIGURE 6.5: Expected topic rank scores for Microblog-hLDA, hLDA, TSSB and rCRP showing that the expected topic rank of hierarchies learned by Microblog-hLDA increase in a smoother fashion with the increase of levels as compared to hLDA, TSSB and rCRP

6.4.1 Heldout Log-Likelihood

For quantitative analysis, we chose the log-likelihood measurement, which is widely used to evaluate how well the trained model explains or predicts held-out data. It is defined as:

$$LogLikelihood = log(p(W_{heldout}|M_{trained})) \tag{6.6}$$

where $W_{heldout}$ is the held-out data and $M_{trained}$ is the trained model [68]. Ten-fold cross-validation was used in all data sets.

We compared held-out log-likelihood of our model to that of hLDA[2], TSSB[3] and rCRP. These models aim to learn topic hierarchies and are non-parametric in terms of numbers of topics, which makes them comparable to our model. Note that rCRP performed significantly worse than other models as it suffered from the chaining effect in our data sets, which we were not able to overcome. The results are visualized in Figure 6.6, which shows that Microblog-hLDA outperforms hLDA, TSSB and rCRP in terms for explanatory power by a significant margin.

[2]http://www.cs.princeton.edu/ blei/downloads/hlda-c.tgz
[3]http://hips.seas.harvard.edu/files/tssb.tgz

6.4.2 Topic Specialization

As discussed in Section 7.3, our intuition is predicated upon an assumption that topics increase in specificity proportionally to their distance from the root of the hierarchy. In our evaluation, we measured the *general-to-specific* characteristic of our model using the *topic specialization* metric introduced by Kim et al. in [68]. The *topic specialization* metric is defined by letting ϕ_{norm} be the baseline topic, such that the probability of generating a word x_i is approximated by the following equation:

$$p(x_i|\phi_{norm}) = \frac{freq(x_i) + \beta}{\sum_{j \in V} freq(x_j) + \beta|V|} \tag{6.7}$$

where $freq(x_i)$ is the frequency of the word x_i in a corpus and V is the vocabulary. That is, ϕ_{norm} is made up of corpus-level proportions for each word and may therefore be considered the most general topic. Then, the distance between topic ϕ_k and ϕ_{norm} is quantified using the cosine distance. Formally, *topic specialization* $\Delta(\phi_k)$ is defined as

$$\Delta(\phi_k) = 1 - \frac{\phi_k \bullet \phi_{norm}}{\parallel \phi_k \parallel \parallel \phi_{norm} \parallel} \tag{6.8}$$

Following the procedure in [68], we averaged *topic specialization* measurements for topics at each level of the hierarchy for Microblog-hLDA, hLDA, TSSB and rCRP. The definition of topic specialization implies that, as topics become more specific, they will drift further away in terms of Δ from the baseline ϕ_{norm}. We, therefore expect that a topic specialization scores will increase linearly in a good topic model hierarchy.

Figure 6.1 summarizes topic specialization scores for Microblog-hLDA, hLDA, TSSB and rCRP. While all evaluated topic modeling techniques appeared to improve in specificity over levels, hLDA, TSSB and rCRP seemed to plateaued quickly. We conjecture that this is likely caused by the noisy and short nature of microblog documents, which lack the content necessary to overcome the preferential attraction of initial assignments during sampling. On the other hand, Microblog-hLDA exhibited a near-linear relationship between levels and specialization. This is expected by design as our model forces specialized terms into appropriate levels.

6.4.3 Expected Topic Rank

While the *topic specialization* score introduced in [68] provides a way to measure the distance between the baseline general topic and learned topics, its usefulness is based on an explicit assumption that the distribution of the baseline topic is near uniform [68, p. 790]. However, since the baseline topic is constructed using word frequencies (see

Equation 6.7), the assumption may not be valid for text corpora, as word frequencies have been shown to follow the Zipfian distribution [74] rather than the uniform one. Therefore, it is theoretically possible for a topic to have high *topic specialization* score according to Equation 6.8, while moving towards the uniform distribution and thus becoming more general.

With that, we introduce a new measure called *expected topic rank* that is the weighted average of ranks in a topic. We use the *expected topic rank* metric to evaluate how rank changes from level to level of learned hierarchies. Let k be a topic index and let ϕ_k be the probability distribution over words for topic k. Then, $R[k]$ is the *expected topic rank* of topic k computed as:

$$R[k] = \sum_{i \in V} \phi_k(x_i) rank(x_i) \tag{6.9}$$

where $rank(x_i)$ is the rank of the word x_i in the frequency table of a given corpus and $\phi_k(x_i)$ is the probability of generating word x_i from topic k.

We note that, since the *expected topic rank* measures average word ranks at each level and since Microblog-hLDA leverages word ranks by construction, our model is expected *apriori* to perform well when measured with *expected topic rank*. The purpose of the *expected topic rank* evaluation in this paper is, therefore, not to validate how well Microblog-hLDA performs, but rather to test how its performance compares to other approaches. That is, since other approaches are expected to learn topic hierarchies that are progressively more specialized towards the leafs [66–68], we aim to test a null hypothesis of no difference between Microblog-hLDA and other models.

We report average *expected topic rank* values for each level of hierarchies learned by Microblog-hLDA, hLDA, TSSB and rCRP for Tweets2011 and IRC data sets in Figure 6.5. We do not report the results for the Twitter 2013 collection to improve readability, as results for that data set are quite similar to those of the Tweets2011 data set. Lower average expected topic rank implies that words in topics of a given level are more general and higher values indicate that words are more specific.

As with the *topic specialization* analysis, *expected topic rank* results for Microblog-hLDA exhibited strong, near-linear relationship between the expected topic rank and hierarchy levels. While other tested approaches also appeared to increase in expected rank with the increase in levels, the pattern for those approaches appeared to plateau quickly, suggesting significant noise in learned hierarchies. Therefore, the null hypothesis of no difference for other models is not supported by our evaluation in terms of the *expected topic rank*.

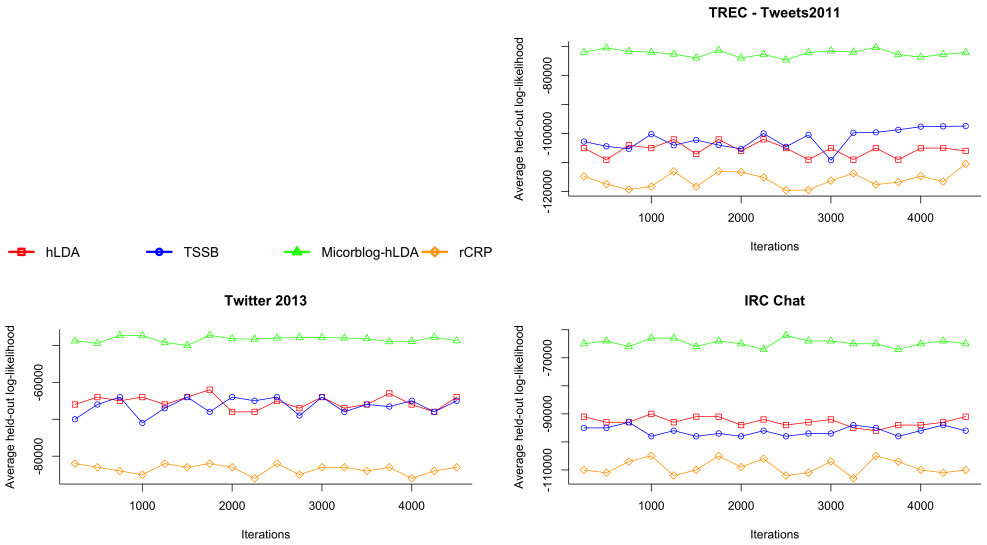


FIGURE 6.6: Held-out log-likelihood for hLDA, TSSB, rCRP and Microblog-hLDA. Higher value of log-likelihood indicates that the model is able to better predict the held-out data.

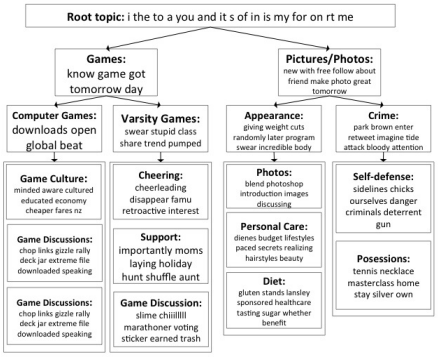


FIGURE 6.7: Comparison of hierarchies inferred by Microblog-hLDA and hLDA. Bold labels are manually chosen to improve readability

6.4.4 Topic Visualization

While log-likelihood, topic specialization and expected topic rank provide quantitative measurements of model performance, qualitative evaluation of differences between hierarchies learned from disjoint social media venues may help induce intuition as to how Microblog-hLDA may be expected to perform in different domains. Unfortunately, it is not possible to show the entire hierarchy of topics in this paper because of space limitations. Therefore, Figure 6.7 presents a representative snippet of hierarchies learned by Microblog-hLDA from the TREC Tweets2011 corpus.

6.5 Conclusions and Future Work

In this paper, we introduced the Microblog-hLDA model that generates inverse power law distributed text corpora and learns hierarchical topic models by leveraging the Zipf's Law property during inference. We applied our model to three distinct data sets and showed that topic models learned by Microblog-hLDA outperformed other approaches in terms of held-out log-likelihood. Then, we tested the topic specialization of topics learned by our model and other approaches. Our evaluation showed that our model produced a near-linear increase in topic specialization, indicating lower levels of noise as compared to TSSB, hLDA and rCRP. Further, we introduced a new metric called *expected topic rank* and showed that Microblog-hLDA exhibited near-linear growth in terms of that metric, which provided further evidence for the improvements in quality of Microblog-hLDA topic model as compared to other tested approaches. Finally, we presented a visualization of a learned topic hierarchy.

The Microblog-hLDA generative process presented in this paper relied on a premise that, if a string is generated by random draws from an alphabet containing a separator character, the resulting collection of tokens, once tokenized, may be proven to follow the inverse power law distribution. While this premise is quite general, it is only applicable for languages where a separator character is in use. Unfortunately, to the best of our knowledge, no proofs are available regarding other human languages, such as Chinese, that do not make use of a separator character. Therefore, in our future work, we will attempt to generalize the Microblog-hLDA process to apply to such languages.

Chapter 7

Learning Focused Hierarchical Topic Models with Semi-Supervision in Microblogs

Topic modeling approaches, such as Latent Dirichlet Allocation (LDA) and Hierarchical LDA (hLDA) have been used extensively to discover topics in various corpora. Unfortunately, these approaches do not perform well when applied to collections of social media posts. Further, these approaches do not allow users to focus topic discovery around subjectively interesting concepts. We propose the new Semi-Supervised Microblog-hLDA (SS-Micro-hLDA) model to discover topic hierarchies in short, noisy microblog documents in a way that allows users to focus topic discovery around interesting areas. We test SS-Micro-hLDA using a large, public collection of Twitter messages and Reddit social blogging site and show that our model outperforms hLDA, Constrained-hLDA, Recursive-rCRP and TSSB in terms of Pointwise Mutual Information (PMI) Score. Further, we test our model in terms of information entropy of held-out data and show that the new approach produces highly focused topic hierarchies.

7.1 Introduction

Modern applications of text mining often deal with large collections of documents that cover diverse sets of topics. Various topic modeling techniques have been developed in recent decades to discover these topics automatically and present visualizations that capture the spectrum of themes in a corpus. In a real-world setting, however, analysts are often interested in grasping the nature of the discourse around a particular concept or entity rather than understanding the corpus as a whole.

Such a task may be difficult to perform when dealing with social media texts. Social microblog systems are populated with millions of noisy, content-poor documents discuss large variety of subjects and concepts. The short, unedited nature of social media texts complicates applications of common topic modeling approaches [60] [61] [62][63][64][65] and makes extraction of interesting patterns especially difficult.

In this paper, we propose the new Semi-Supervised Microblog-hLDA (SS-Micro-hLDA) model that learns topics (defined as probability distributions over words) from microblog data in a way that allows for sets of interesting keywords (referred to as *supervisory word sets* from here on) to influence the topic learning process. To make the job of interpreting the learned topics easier, we require our approach to organize topics as hierarchies. This is motivated by well-known works in cognitive research that suggest that hierarchies may be instrumental in enhancing human sense making [1, 2].

We test the new approach using the standard Tweets2011 data set made public by the TREC project and show that our model produces more interpretable and coherent topic models when measured in terms of PMI-Score against TSSB, Recursive-CRP, Constrained-hLDA and hLDA. Further, we test our approach and related approaches using information entropy and show that our model learns topic hierarchies that are more subject-focused than those produced by TSSB, Recursive-CRP, Constrained-hLDA and hLDA.

The paper is organized as follows. Section 7.2 discusses the current state of research in the area of topic modeling in general and microblog topic modeling in particular. Section 7.3 offers an analysis of topic modeling challenges in social stream data and describes the new Semi-Supervised Microblog-hLDA model designed to overcome these challenges. In Section 7.4, we discuss data sets and experiments that were used to evaluate how well our new topic modeling approach performed as compared with other approaches. Section 7.5 concludes the paper and outlines future work.

7.2 Related Works

Discovering hidden relationships between words may be accomplished using a number of different techniques. Matrix factorization approaches such as Latent Semantic Indexing (LSI) [76] and Non-Negative Matrix Factorization (NMF) [77] have been used to infer latent relationships between terms. While matrix factorization may be employed for topic discovery, approaches based on Latent Dirichlet Allocation (LDA) [69] have become very popular in recent years. This popularity has often been attributed to the flexibility

and modularity of LDA, which easily lends itself to extensions and generalizations that accommodate many types of relationships in data [70].

LDA is a generative probabilistic model that makes the "Bag-of-Words" assumption and represents documents as probability distributions over K topics. These topics are, in turn, viewed as probability distributions over W words.

While LDA has enjoyed much popularity serving as basis for numerous extensions and generalizations, one of its major limitations is that users must select the number of topics K before the approach can be used. This requirement makes the approach quite rigid, as it cannot accommodate influx of new data [66]. To make topic modeling more flexible, LDA machinery was modified in [66] to use the Chinese Restaurant Process (CRP) [71]. CRP relaxes the fixed K constraint of LDA by assuming an infinite number of topics and postulating that words are generated from topics chosen according to the following distribution:

$$\begin{aligned} p(\textit{existing topic } i|\textit{previous words}) &= \frac{m_i}{\lambda + m - 1} \\ p(\textit{new topic}|\textit{previous words}) &= \frac{\lambda}{\lambda + m - 1} \end{aligned} \tag{7.1}$$

where m_i is the number of words assigned to topic i, λ is a parameter and m is the total number of words seen so far. The formulation in Equation 7.1 removes the need to know K *apriori* as it assigns a non-zero probability to choosing a new topic. This allows the number of discovered topics to grow as new data arrives.

To improve interpretability of discovered topics, the work by Blei et al. on Hierarchical LDA (hLDA)[66] attempted to learn organized topic hierarchies. The hLDA generative probabilistic model assumes that words in a document are generated from an infinitely branched tree of height L according to a document-specific mixture model. In hLDA, each node of the tree is associated with a single topic.

To learn topic and tree structure from data, sampling is often used by first choosing an L-level path c_d for each document d according to Equation 7.2.

$$p(\boldsymbol{c}_d|\boldsymbol{w}, \boldsymbol{c}_{-d}, \boldsymbol{z}) \propto p(\boldsymbol{w}_d|\boldsymbol{c}, \boldsymbol{w}_{-d}, \boldsymbol{z}) p(\boldsymbol{c}_d|\boldsymbol{c}_{-d}) \tag{7.2}$$

where $\boldsymbol{w}_{-d}$ and $\boldsymbol{c}_{-d}$ are words and paths of documents other than d; $\boldsymbol{w}$ and $\boldsymbol{c}$ are respectively words and paths of all documents; $\boldsymbol{z}$ is the topic assignments. Once the path is found, topic assignments for words are approximated by sampling [60].

Including partial supervision in topic modeling has been an active area of research in recent years. Approaches such as the one proposed in [78] and extended in [79] work by defining sets of possible topic assignments for each word type (referred to as *topic-in-sets*)

and modifying the Gibbs sampler to constrain possible topic choices for words according to these *topic-in-sets*. These approaches are flexible in that they allow certain "seed words" to help focus topic discovery for words and documents from underrepresented or noisy topics. Unfortunately, specifying *topic-in-sets* early in the topic discovery process is only meaningful if the number of topics is known ahead of time. This limits the usefulness of these approaches as they cannot be applied in a non-parametric settings, such as the nested CRP or the Hierarchical LDA.

Semi-Supervised hLDA (SSHLDA) proposed in [80] introduces partial supervision into the Hierarchical LDA learning process by restricting the initial structure of the topic tree to known hierarchies of labels and then allowing the nested CRP process to discover new branches in the tree with stochastic sampling, as in hLDA.

Constrained-hLDA is particularly relevant to this work because, as in this paper, it focuses on hierarchical topic modeling in microblogs. Specifically, Constrained-hLDA experimented with Chinese microblogs and showed significant improvement in terms of held-out log-likelihood. As noted by the authors, much of the improvements were realized by an additional heuristic aimed specifically at microblog data, which restricted word-level assignments during sampling. Their relied on the document frequency function, which returned the number of documents containing a word in a corpus, as well as upper and lower inclusion boundary thresholds and part-of-speech indicators.

The novel model discussed in the next section improves upon *topic-in-set*-based approaches, such as the one proposed in [78], by allowing for partial supervision and guidance to be applied to hierarchical topic learning in a way that is non-parametric with respect to the number of topics. The new model further improves on recently proposed hierarchical semi-supervised approaches in that it incorporates supervision in a way that does not require an existing label hierarchy (as in SSHLDA) nor does it necessitate the initial supervisory hierarchy to be learned by other means (such as FP-Tree in Constrained-hLDA).

7.3 Semi-Supervised Microblog-hLDA Model

We motivate our model by imagining that topics are not atomic constructs, but are rather comprised of levels of topic specificity. That is, we consider that microblog posts pertain to a single conceptual theme (such as the presidential election or the World Cup), and that each theme contains a number of stages or levels of specificity. For example, when discussing the World Cup, one microblog message may express excitement about the fact of the World Cup's existence, while another post may speak about an outcome

of a particular match or the role of a specific player. In both cases, messages may be set to belong to a "World Cup" theme, but the former message is surely more general than the latter one.

The above intuition may be captured by making a simple modification to hLDA to sample levels according to a uniform distribution, rather than a multinomial one.

7.3.1 Generative Process for Semi-Supervision

We, then, take our approach a step further and attempt to discover a way to allow semi-supervised focus to be introduced into topic modeling. That is, we imagine a user interested in a particular subject area supplies a topic modeling algorithm with few keywords or phrases about the subject. The user, then, expects the algorithm to highlight her keywords and phrases by restricting them to a single position in the resulting topic tree. Further, the user may expect the approach to discover topics *around* the given subject area (siblings, parents, etc.) providing the user with further insights into her area of interest.

The novel approach, which we term Semi-Supervised Microblog hLDA, is outlined in Figure 7.1. It assumes that for each social media collection, there exists a parallel corpus of short phrases, which has a bearing on how microblog posts are generation. We treat the parallel corpus as a collection of word phrases and refer to these phrases as *supervisory word-sets*. We, then, imagine that these supervisory word sets are themselves generated with a random generative process.

Figure 7.1 depicts the resulting algorithm. There, $W_{sup} = \{\mathbf{w}_1, ..., \mathbf{w}_S\}$ is a collection of supervisory word-sets, such that $\mathbf{w}_s \in W_{sup} = \{w | w \in V\}$, and $S = |W_{sup}|$ is the number of supervisory word-sets. The process starts by generating S supervisory word-sets in step 3. In step 3(c)c, supervisory words are aggregated into the set $\mathbf{W}_{sup}$, which is used in later steps to ensure that supervisory words may only be generated on paths associated with supervisory word sets. In step 3d, leaf nodes of paths chosen for each of the supervision word sets are aggregated into a set L_{sup}. The resulting collection of S paths is used in later steps to ensure that words in supervisory sets may only emerge from paths associated with those supervisory sets.

Having generated the supervisory sets, the process begins to produce document content in step 4. First, the process draws a number $x \in \{1, ..., L\}$ from a multinomial distribution parameterized by an L-sized vector σ (step 4a). Then, an index s into the set L_{sup} is drawn from a distribution parameterized by an $|L_{sup}|$-dimensional vector w. Then, the process chooses a path for each document by deterministically selecting the first x

1. Let $L_{sup} = \emptyset$ be a collection of paths
2. Let $\mathbf{W}_{sup} = \emptyset$ be a collection of words
3. For supervisory word-set $\mathbf{w}_s \in W_{sup}$
   a. Let c_1 be the root
   b. For each level $l \in \{2, \ldots, L\}$:
      a) Draw a child node form c_{l-1} using Equation 7.1. Set c_l to be that node
   c. For each word $n_{sup} \in \{1, \ldots, |\mathbf{w}_s|\}$:
      a) Draw $z \in \{1, \ldots, L\}$ from $Uniform(L)$
      b) Draw w from the topic associated with c_z
      c) Set $\mathbf{W}_{sup} = \mathbf{W}_{sup} \cup \{w\}$
   d. Set $L_{sup} = L_{sup} \cup \{c_l\}$
4. For each document
   a. Draw $x \in \{1, ..., L\}$ from $Mult(\sigma)$
   b. Draw $s \in \{1, ..., |L_{sup}|\}$ from $Mult(\omega)$
   c. Select s^{th} node c_s from L_{sup}
   d. Let c_1 be the root
   e. For each level $l \in \{2, \ldots, x\}$:
      a) Select the l^{th} node c_l from the path to node c_s
   f. For each level $l \in \{x+1, \ldots, L\}$:
      a) Draw a child node form c_{l-1} using Equation 7.1. Set c_l to be that node
   g. For each word $n \in \{1, \ldots, N\}$:
      a) Draw $z \in \{1, \ldots, L\}$ from $Uniform(L)$
      b) Let β_{c_z} be the word-topic proportions vector associated with node c_z
      c) If($z < x$):
         1. Construct set $V' = (V \backslash \mathbf{W}_{sup}) \cup \mathbf{w}_s$
      d) If($z >= x$):
         1. Construct set $V' = V \backslash \mathbf{W}_{sup}$
      e) Construct a $|V| \times |V|$ diagonal matrix M^s such that for each $i = j$, $M^s_{ij} = cell(i, V')$ (see Equation 7.3)
      f) Let $\beta'_{c_z} = M^s \times (\beta_{c_z})$
      g) Draw $w \in V$ from $Mult(\beta'_{c_z})$

FIGURE 7.1: Semi-Supervised Microblog-hLDA generative process

nodes from the s^{th} path in L_{sup} (step 4e) and then allowing the CRP to randomly choose nodes from $(s+1)^{th}$ level to the leaf level L (step 4f). It is important to note that the realization $x = 1$ in step 4a amounts to no supervision, since all paths share the root node.

The resulting path is used to generate words in the document. For each word, the process draws $z \in \{1, \ldots, L\}$ from a uniform distribution. Once the level assignment is known, set V' is constructed in step 4(g)c and initially contains all words in vocabulary V except for all supervisory terms of set $\mathbf{W}_{sup}$. If the chosen node assignment is on the path associated with some supervisory word-set, the supervisory words of that set are added to V'. Then, the $|V|$-dimensional word proportions vector associated with the chosen topic multiplies a diagonal matrix, which contains zeros in elements of the diagonal that correspond to indices of words not found in vector V'. The multiplication in step 4(g)f has the effect of allowing supervisory words to be generated only from a single hierarchy path. The resulting unnormalized parameter vector is used to randomly select words in a way identical to hLDA.

$$cell(i, V) = \begin{cases} 1 & \text{if } w_i \in V \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

7.3.2 Inference

Posterior inference tries to visualize hidden process structures by repeatedly adjusting its mental vision of them to better fit the actual observations. In our application, observations consist of two collections – 1) a corpus of microblog messages and, 2) a number of user-specified supervisory word sets. Given these observations, we are interested in learning the shape of the hidden topic hierarchy and the word proportions for the nodes in the hierarchy.

With that, the posterior inference for our model is conducted as follows. First, a random tree scaffolding is constructed by many hierarchical random walks. Then, for each microblog message, an L-level path through the scaffolding is selected. This is followed by a path node selection and subsequent counters updated for both path and the node. These counters are used in later stages to approximate hierarchy makeup and parameters.

The L-level path is selected according to Equation 7.2. Once the path is chosen, the algorithm knows that all words in the message were generated by the particular path, but still needs to determine which member of the path was responsible for which words. This is a challenge as our model postulates a uniform distribution over levels for each

document, which means that the posterior inference algorithm cannot learn level assignments from data, as in hLDA. To have a reasonable chance of intelligently approximating the hidden structure, the inference procedure may consider the following argument.

The uniform distribution postulate of our approach implies that each document draws equally many words from each of the levels, but gives no guidance as to how to determine which level of the hierarchy generated which of the words. If the hidden word distributions at each level were known, the inference algorithm could simply choose a node with the highest probability of a given word. However, since these distributions are unknown, the inference procedure may consider the following dichotomy regarding word proportions in nodes of the hidden tree – 1) all distributions are of the same (or similar) shape and, 2) all distributions are **not** of the same (or similar) shape.

The notion that all the distributions are the same or similar contradicts with everyday common sense – obviously, language texts, such as social media posts, discuss a variety of subjects. Therefore, we must conclude that words are distributed unequally among paths and, consequently, path nodes. With that, words that are favored by 'popular' nodes (nodes that appear on many paths) must appear more frequently than words from unpopular nodes. Again arguing from the observations, because empirical laws (e.g.: the Zipf's Law [74]) suggest that words are distributed according to the inverse power-law, there must be few 'popular' nodes and many 'unpopular' ones. Then, in graph-theoretic terms, since, by definition, there are fewer higher-level nodes than lower-level ones, the higher-level nodes (those closer to the root) must be the 'popular' ones and the lower-level (towards leafs) nodes must be relatively 'unpopular'.

With that, the level assignment task is straight forward. Given a word, the algorithm may simply consult a frequency table and determine its corpus-level rank. Then, if the word ranks first, the word must be associated with the root node, whereas if its ranked last, it gets assigned to the leaf node of the given document path.

Naturally, the above raises the question of what to do if the rank is somewhere between first and last. We tackle this challenge by partitioning the corpus frequency table into L *buckets* (one for each hierarchy level) in such a way as to place few highly ranked words into the top-level bucket and many very infrequent terms into leave level one. This intuition is quantified by assigning words to levels during sampling according to the following equation:

$$Level_w = \lfloor log_{\sqrt[L]{N+1}}(rank(w)) \rfloor + 1 \tag{7.4}$$

where N is the number of distinct words and $rank(w)$ is the rank of word w in the corpus frequency table. The equation captures our intuition by exponentially increasing bucket sizes towards the leaf level.

For an illustrative example, when considering a 3-level hierarchy ($L = 3$) and a 1000 term vocabulary ($|V| = 1000$), Equation 7.4 will associate the 10 most frequent terms with the root level, next 90 with the intermediate level, and 900 least frequent ones with the leaf nodes. Then, for the same hierarchy, if the word *"the"* were the most frequent word in a corpus containing 1000 unique terms, its rank would necessarily be 1 and $Level_{"the"} = \lfloor log_{\sqrt[3]{1001}}(1) \rfloor + 1 = 1$, which is the root level. If, however, the word *"unique"* were the only word to appear just once in the corpus, it would be ranked 1000 and its level would be computed as $Level_{"unique"} = \lfloor log_{\sqrt[3]{1001}}(1000) \rfloor + 1 = 3$, which is the leaf.

During inference, we approximate each word's position with the value of $Level_w$ for each observed word w by deterministically selecting level assignments with the help of Equation 7.4.

7.3.3 Inference with Semi-Supervision

We outline the supervised inference procedure by recalling that, in addition to a document corpus, observations in the SS-Micro-hLDA also contain collections of supervisory words. Since SS-Micro-hLDA uses the same generative approach for both the supervisory and the document corpora, same sampling procedure may apply. The restriction that supervisory words may originate from only a single path (step 4(g)f in Figure 7.1) implies that documents containing supervisory words must have been generated from paths that share a prefix with paths to leafs associated with supervisory word sets.

To introduce supervision into the sampling process, we start by randomly and without replacement selecting a node from a set of hierarchy leafs for each supervisory set $\mathbf{w}_s \in W_{sup}$. This results in a collection of tuples $\mathbf{S}_{sup} = \{< \mathbf{w}_1, c_1 >, ..., < \mathbf{w}_S, c_S >\}$ such that each c_i is a leaf node and $|\mathbf{S}_{sup}| = |W_{sup}|$. Then, for each i^{th} tuple $S_i \in \mathbf{S}_{sup}$, words in its word set $\mathbf{w}_i$ are assigned to nodes on the path to c_i according to word ranks as specified by Equation 7.4.

Then, for each observed document, topic hierarchy path is selected by first checking whether any words in the document are found in any supervisory set and constraining the path selection to go through the corresponding node. Once, the path is known, word assignments are sampled according to Equation 7.4.

7.4 Evaluation

The proposed model was tested with two datasets – the Tweets2011 Twitter Collection made available through the TREC project [75] and a collection of user comments on a popular Reddit news and social networking site, which we manually collected by monitoring the site's programmatic API end-points. The Twitter data set consisted of 16 million Twitter messages sampled in early months of 2011. The Reddit collection was comprised of $51,563$ user comments to articles posted in Reddit subsections (known as *subreddits*) labeled */gaming*, */politics* and */sports*.

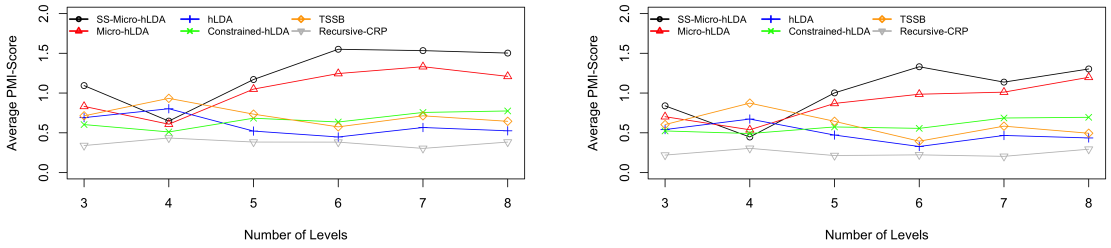
It is common knowledge that many social media messages are tagged with special topical annotations known as *hashtags*. While users often misplace or misspell hashtags or abuse the hashtag notation (i.e.: some messages may contain more hashtags than actual text), with no standard corpus available, our approach was tested on collections of carefully select tagged messages.

To construct a test corpus, we parsed the English language messages in the Tweets2011 collection and assembled corpus-level hashtag counts. We then selected those hashtags that appeared in at least 1000 messages in the corpus. The resulting 34 hashtags were used to construct the corpus by retaining only those messages that contained the frequent tags. The data set was further restricted to those Twitter messages that contained only a single hashtag. This was done to control noise with the intuition that messages with just a single hashtag are more likely to be focused on a particular subject.

7.4.1 PMI-Score Evaluation

To compare performance of our approach to others, we expressed our interest in the Egyptian revolution and the major American Football sporting event by constructing two supervisory sets – {*'protests', 'egypt'*} and {*'super', 'bowl','packers','steelers'*}. We then trained topic models using semi-supervised and unsupervised variants of our approach[1] as well as the Constrained-hLDA and hLDA (to serve as a baseline) and compared resulting models in terms of the PMI-Score [81]. The PMI-Score measure was chosen in favor of other metrics, such as such as perplexity or log-likelihood, as this measure has been reported by numerous researchers ([82],[81],[83]) to correlate well with human interpretation of topic models.

[1] Unsupervised variant of SS-Micro-hLDA is achieved trivially by providing an empty collection of supervisory word sets



(a) Twitter2011 (b) Reddit

FIGURE 7.2: Average PMI-Score evaluation results

PMI-Score is motivate by the observation that human evaluation of topic models is often conducted by considering the top n representative words for each topic. The PMI-Score aims to provide quantitative approximation of human evaluation by considering the Pointwise Mutual Information for the top n words as quantified by Equation 7.5.

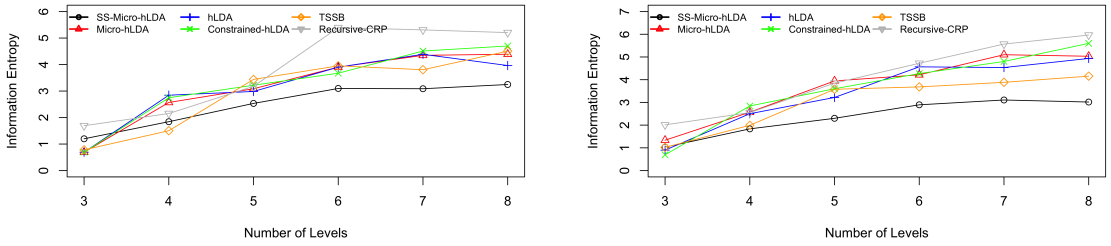
$$PMI - Score(\mathbf{w}) = median\{PMI(w_i, w_j), ij \in \{1, ..., n\}\} \tag{7.5}$$

where $\mathbf{w}$ is the topic, w_i and w_j are i^{th} and j^{th} ranked words in topic $\mathbf{w}$, n is the number of 'top words' selected (for example, n=10 top words), $PMI(w_i, w_j) = \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$.[81]

Evaluation results using ten-fold cross-validation are outlined in Figure 7.2. The figure reports PMI-Scores for hierarchies of different heights and shows that SS-Micro-hLDA outperforms other approaches for deeper hierarchies. All models appeared to perform similarly in terms of the PMI-Score for shallower hierarchies (number of levels less than 5). This is expected as shallow hierarchies do not allow for deep specialization in topic structures.

7.4.2 Information Entropy Evaluation

While the PMI-Score evaluation presented above tested topic models in terms of their interpretability, the metric did not measure how well sections of hierarchies focused on particular topical areas. That is, in hierarchical topic learning, it is expected that siblings are somehow conceptually related to one another. For example, topics on "dogs" and "cats" may be expected to appear under the general topic on "mammals", while "apples" and "oranges" should occur under the general topic heading on "fruits". If a hierarchical topic modeling approach were to place the "dogs" topic under the "fruits" heading, a human analyst would likely find such a placement in error even if the top words of the topic were coherent and interpretable.



(a) Twitter – *#egypt* (b) Reddit – */sports*

FIGURE 7.3: Entropy results for Twitter #egypt and #superbowl and Reddit /sports and /politics data

To evaluate how closely the model places related documents, we estimated probabilities of each node as proportional to the number of times a node appeared on any document's path. We then computed Shannon's information entropy [84] given as $H(C) = -\sum_d p(c_d|testdata)log(p(c_d|testdata))$ where $C = \{c_1, ..., c_{N_{test}}\}$ is a random variable taking on values of all possible paths. The information entropy quantity may be interpreted by considering that, in a *focused* hierarchy, test documents on the same topic would likely be concentrated in a particular area of the hierarchy, their placement being more predictable and implying lower entropy. On the other hand, classification using an *unfocused* hierarchical model would place documents more evenly across the entire hierarchy, resulting in higher entropy. Therefore, we would expect the information entropy of a focused hierarchy to be lower than that of an unfocused one.

Results of the information entropy evaluation are presented in Figure 7.3. We only present results for the Twitter *#egypt* and Reddit *#sports* test samples because of space considerations. In Figure 7.3, information entropy for the test data using SS-Micro-HLDA model is lower than that of other models for deeper hierarchies, suggesting a more focused topic tree. This is particularly encouraging as deeper hierarchies provide a way for analysts to focus on a particular area among a potentially large number of topics.

7.5 Conclusions and Future Work

In this paper, we developed an algorithm to infer hierarchical topic models around specific concepts that may be of interest to analysts. We evaluated our new algorithm using a large, publicly available collection of microblog messages and showed that the proposed method outperformed other approaches in terms of the PMI-Score. As PMI-Score has

been shown to relate favorably to topic interpretability by humans, this evaluation suggests that our new approach produces highly meaningful topic models.

While we were able to show that our new approach preforms better than existing state-of-the-art topic modeling on a static data set, our approach is not designed for continuous operation on stream data. In our future work, we will focus on developing an approach to handle streaming social media messages with the goal of tracking and monitoring social discourse over time.

Chapter 8

Conclusions and Future Work

Automatic discovery of valuable insights in social media is becoming a very relevant challenge for both the academia and the industry. New, emerging social media outlets such as Snapchat, Kik and others are empowering users with new capabilities and novel ways to conduct social discourse. The plethora of various types of content produced by popular social venues is sure to contain interesting research artifacts as well as monetizable business value.

Therefore, this thesis attempted to make strides towards developing meaningful and effective data mining strategies for efficiently discovering new knowledge in social streams. The thesis began in Chapter 2 by considering a narrow and focus challenge of making sense of data which, akin to social media utterings, contained by a few distinct words and was noisy and unstructured. It then improved upon the initial approach in Chapter 3 by considering the data mining effort in terms of a hierarchical scaffolding and introduced a novel algorithm to represent short and noisy data as corresponding to nodes in a well-organized, meaningful concept taxonomy. This approach we immediately found to be useful in practice in [37]. In Chapter 4, I began to tackle more general data and applied hierarchical topic discovery approaches to a collection of Web reviews, which are, again, similar in their nature to social media stream data in terms of noise and brevity of content.

The algorithm presented in Chapter 5 made strides towards answering my first research question of "How to conduct topic discovery in social streams in a scalable way while improving quality of topic modeling'?' by applying lessons learned in my earlier efforts to social media data collected from Twitter and other popular systems. In that chapter, I advanced further towards efficiently mining social streams by developing an approach for scalable hierarchical topic modeling in microblogs. The approach overcame many

scalability challenges of current approaches and produced measurably better results in terms of quality of topics found and hierarchical relationships discovered.

In Chapters 6 and 7, the second research question of "How can concept graphs be used to represent social discourse in microblogs?" was tackled by re-examining some of the basic assumptions of existing topic modeling frameworks and applying semi-parametric learning to improve performance. The resulting approach outperformed modern state-of-the-art topic mining techniques and produced hierarchical visualizations of social streams that were more interpretable and meaningful compared to other approaches in terms of the PMI-Score.

8.1 Future Work

It is clear that the next step to further enhance knowledge discovery in microblogs is to combine the algorithms presented in Chapters 5,6 and 7 into a single system that would not only discover meaningful hierarchical structures, but also operate in a scalable manner. While such a system may be a simple combination of my earlier efforts and may not amount to a scientifically interesting publication, it seems it may be of help to solve real-world challenges for analysts and researchers. In my future work, I will continue working on improving knowledge discovery in social stream as I feel this task is essential in the modern world.

Appendix A

Percent error for frequent titles

Here, Classification Baseline, Extended and Manual columns contain SOC labels for corresponding text entry in the Title column assigned by the baseline algorithm, algorithm proposed in this paper and manual classifier respectively. Physical demand score Baseline, Extend and Manual columns contain physical demand scores associated with SOC labels assigned by the baseline algorithm, algorithm proposed in this paper and human classifiers respectively. Percent error columns contain percent difference between physical demand values associated with SOC labels assigned manually and values associated with SOC labels assigned by the baseline algorithm and those values assigned by the algorithm proposed here respectively.

Title	**Count**	**Baseline**	**Extended**	**Manual**
teacher	2709	25-2011	25-2000	25-2000
secretary	517	43-6013	43-6010	43-6010
teaching	472	25-2011	25-2000	25-2000
waitress	370	35-3031	35-3030	35-3031
nurse	217	29-1172	29-1172	29-1141
school teacher	201	25-2031	25-2000	25-2000
rn	167	29-1141	29-1140	29-1141
social worker	152	21-1022	21-1020	21-1020
librarian	145	25-4021	25-4020	25-4021
telephone operator	116	43-2011	43-2010	43-2021
sales	114	Nov-22	Nov-22	41-0000
registered nurse	111	29-1141	29-1140	29-1141
clerk	106	43-4161	43-0060	43-9061
nursing	102	29-1172	29-1172	29-1141

Table A.1: Classification Results

Title	**Count**	**Baseline**	**Extended**	**Manual**
teacher	2709	44.8	35.8	35.8
secretary	517	24.66	25.725	25.73
teaching	472	44.8	35.8	35.8
waitress	370	55.17	55.17	55.17
nurse	217	35	35	45.24
school teacher	201	30.34	35.8	35.8
rn	167	45.244	45.24	45.24
social worker	152	30.53	30.6	30.6
librarian	145	28.11	28.11	28.11
telephone operator	116	25.69	25.693	22.96
sales	114	24.66	24.66	32.1
registered nurse	111	45.24	45.24	45.24
clerk	106	24.8	24.8	29.3
nursing	102	34.8	34.98	45.24

Table A.2: Physical Demand Results

Title	**Count**	**Baseline**	**Extended**
teacher	2709	25	0
secretary	517	4	0
teaching	472	25	0
waitress	370	0	0
nurse	217	22.69	22.69
school teacher	201	15.24	0
rn	167	0	0
social worker	152	22	0
librarian	145	0	0
telephone operator	116	12	12
sales	114	23	23
registered nurse	111	0	0
clerk	106	15	15
nursing	102	22.7	22.7

Table A.3: Percent error relative to manual physical demand based on manual code

Bibliography

[1] Carol Conrad. Cognitive economy in semantic memory. 1972.

[2] Allan M. Collins and Elizabeth F. Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975.

[3] citeulike:9335608. Tweets2011 corpus. Online, 2011. URL https://sites.google.com/site/microblogtrack/2011-guidelines.

[4] AGENCY FOR TOXIC SUBSTANCES and DISEASE REGISTRY (ATSDR). Toxicological profile for beryllium. u.s. department of health and human services, public health service. 2002.

[5] NATIONAL INSTITUTE FOR OCCUPATIONAL SAFETY, HEALTH (NIOSH), U.S.DEPARTMENT OF HEALTH EDUCATION, and WELFARE (DHHS). National occupational exposure survey sampling methodology. 1990.

[6] Jan Komorowski, Zdzislaw Pawlak, Lech Polkowski, and Andrzej Skowron. Rough sets: A tutorial, 1998.

[7] Michele P. Hamm and Igor Burstyn. Estimating occupational beryllium exposure from compliance monitoring data. *Archives of Environmental & Occupational Health*, 66(2):75–86, 2011. doi: 10.1080/19338244.2010.511309. URL http://dx.doi.org/10.1080/19338244.2010.511309. PMID: 24484364.

[8] T Vaughan, P Stewart, K Teschke, C Lynch, G Swanson, J Lyon, and M Berwick. Occupational exposure to formaldehyde and wood dust and nasopharyngeal carcinoma. *Occupational and Environmental Medicine*, 57(6):376–384, 06 2000. doi: 10.1136/oem.57.6.376. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1739963/.

[9] Jérôme Lavoué, Raymond Vincent, and Michel Gérin. Formaldehyde exposure in u.s. industries from osha air sampling data. *Journal of Occupational and Environmental Hygiene*, 5(9):575–587, 2015/03/08 2008. doi: 10.1080/15459620802275023. URL http://dx.doi.org/10.1080/15459620802275023.

[10] G.K. Zipf. *Selected Studies of the Principle of Relative Frequency in Language, by George Kingsley Zipf,...* Harvard University Press, 1932. URL http://books.google.com/books?id=Brp2XwAACAAJ.

[11] Grzegorz Kondrak. N-gram similarity and distance. In *Proc. Twelfth Int'l Conf. on String Processing and Information Retrieval*, pages 115–126, 2005.

[12] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September 1999. ISSN 0360-0300. doi: 10.1145/331499.331504. URL http://doi.acm.org/10.1145/331499.331504.

[13] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *In Proc.ofthe15thInt.Conf.onDataEngineering*, 2000.

[14] George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling, 1999.

[15] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5. URL http://dl.acm.org/citation.cfm?id=275537.275705.

[16] Daniel Crabtree, Xiaoying Gao, and Peter Andreae. Standardized evaluation method for web clustering results. In *2005 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2005), 19-22 September 2005, Compiegne, France*, pages 280–283, 2005. doi: 10.1109/WI.2005.138. URL http://dx.doi.org/10.1109/WI.2005.138.

[17] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[18] Gamila Obadi, Pavla Drázdilová, Lukas Hlavacek, Jan Martinovic, and Václav Snásel. A tolerance rough set based overlapping clustering for the dblp data. In *Web Intelligence/IAT Workshops*, pages 57–60. IEEE, 2010. URL http://dblp.uni-trier.de/db/conf/iat/iatw2010.html#ObadiDHMS10.

[19] Pradeep Kumar, P. Radha Krishna, Raju S. Bapi, and Supriya Kumar De. Rough clustering of sequential data. *Data Knowl. Eng.*, 63(2):183–199, 2007. URL http://dblp.uni-trier.de/db/journals/dke/dke63.html#KumarKBD07.

[20] Ning Shan, Wojciech Ziarko, Howard J. Hamilton, and Nick Cercone. Using rough sets as tools for knowledge discovery. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada,*

August 20-21, 1995, pages 263–268, 1995. URL http://www.aaai.org/Library/KDD/1995/kdd95-046.php.

[21] Duo Chen, Du wu Cui, Chao xue Wang, and Zhu rong Wang. A rough set-based hierarchical clustering algorithm for categorical data. *International Journal of Information Technology*, 12:149–159, 2006.

[22] B. o. L. Statistics. http://www.bls.gov/soc/.

[23] Jennifer Hays, Julie R Hunt, F Allan Hubbell, Garnet L Anderson, Marian Limacher, Catherine Allen, and Jacques E Rossouw. The women's health initiative recruitment methods and results. *Ann Epidemiol*, 13(9 Suppl):S18–77, Oct 2003. ISSN 1047-2797 (Print); 1047-2797 (Linking).

[24] Design of the women's health initiative clinical trial and observational study. the women's health initiative study group. *Control Clin Trials*, 19(1):61–109, Feb 1998. ISSN 0197-2456 (Print); 0197-2456 (Linking).

[25] D.O.H.A.H. Services: and N.I.O. Safe. *National Occupational Exposure Survey Sampling Methodology*. BiblioBazaar, 2013. ISBN 9781288666287. URL http://books.google.com/books?id=YpYwmwEACAAJ.

[26] Heung-Seon Oh, Yoonjung Choi, and Sung-Hyon Myaeng. Combining global and local information for enhanced deep classification. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1760–1767, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-639-7. doi: 10.1145/1774088.1774463. URL http://doi.acm.org/10.1145/1774088.1774463.

[27] Heung-Seon Oh, Yoonjung Choi, and Sung-Hyon Myaeng. Text classification for a large-scale taxonomy using dynamically mixed local and global models for a node. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 7–18, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. URL http://dl.acm.org/citation.cfm?id=1996889.1996895.

[28] Andrei Z. Broder, Marcus Fontoura, Evgeniy Gabrilovich, Amruta Joshi, Vanja Josifovski, and Tong Zhang. Robust classification of rare queries using web knowledge. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 231–238, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277783. URL http://doi.acm.org/10.1145/1277741.1277783.

[29] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM COMPUTING SURVEYS*, 34:1–47, 2002.

[30] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 521–528, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1119-8. URL http://dl.acm.org/citation.cfm?id=645496.657884.

[31] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 78–87, New York, NY, USA, 2004. ACM. ISBN 1-58113-874-1. doi: 10.1145/1031171.1031186. URL http://doi.acm.org/10.1145/1031171.1031186.

[32] M. Sasaki and K. Kita. Rule-based text categorization using hierarchical categories. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 3, pages 2827–2830 vol.3, Oct 1998. doi: 10.1109/ICSMC.1998.725090.

[33] Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, 7(1):36–43, June 2005. ISSN 1931-0145. doi: 10.1145/1089815.1089821. URL http://doi.acm.org/10.1145/1089815.1089821.

[34] URL http://www.bls.gov/soc/soc_2010_direct_match_title_file.pdf.

[35] Yvonne L Michael, Nichole E Carlson, Rowan T Chlebowski, Mikel Aickin, Karen L Weihs, Judith K Ockene, Deborah J Bowen, and Cheryl Ritenbaugh. Influence of stressors on breast cancer incidence in the women's health initiative. *Health psychology : official journal of the Division of Health Psychology, American Psychological Association*, 28(2):137–146, 03 2009. doi: 10.1037/a0012982. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2657917/.

[36] URL http://www.onetonline.org/.

[37] Igor Burstyn, Anton Slutsky, Derrick G Lee, Alison B Singer, Yuan An, and Yvonne L Michael. Beyond crosswalks: reliability of exposure assessment following automated coding of free-text job descriptions for occupational epidemiology. *Ann Occup Hyg*, 58(4):482–492, May 2014. ISSN 1475-3162 (Electronic); 0003-4878 (Linking). doi: 10.1093/annhyg/meu006.

[38] URL http://www.dmoz.org/.

[39] David M. Blei and Jon D. Mcauliffe. Supervised topic models. In *In preparation*. MIT Press, 2008.

[40] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora.

In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL http://dl.acm.org/citation.cfm?id=1699510.1699543.

[41] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937.

[42] Caimei Lu, Xiaohua Hu, Xin Chen, Jung-Ran Park, TingTing He, and Zhoujun Li. The topic-perspective model for social tagging systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 683–692, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835891. URL http://doi.acm.org/10.1145/1835804.1835891.

[43] Adler J. Perotte, Frank Wood, Noemie Elhadad, and Nicholas Bartlett. Hierarchically supervised latent dirichlet allocation. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 2609–2617, 2011. URL http://dblp.uni-trier.de/db/conf/nips/nips2011.html#PerotteWEB11.

[44] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. URL http://scholar.google.de/scholar.bib?q=info:NVEeNb3JVywJ:scholar.google.com/&output=citation&hl=de&ct=citation&cd=0.

[45] Yves Petinot, Kathleen McKeown, and Kapil Thadani. A hierarchical model of web summaries. In *ACL (Short Papers)*, pages 670–675. The Association for Computer Linguistics, 2011. ISBN 978-1-932432-88-6. URL http://dblp.uni-trier.de/db/conf/acl/acl2011s.html#PetinotMT11.

[46] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM. ISBN 0-89791-962-9. doi: 10.1145/276698.276876. URL http://doi.acm.org/10.1145/276698.276876.

[47] L. AlSumait, D. Barbara, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 3–12, Dec 2008. doi: 10.1109/ICDM.2008.140.

[48] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557121. URL http://doi.acm.org/10.1145/1557019.1557121.

[49] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0217-3. doi: 10.1145/1964858.1964870. URL http://doi.acm.org/10.1145/1964858.1964870.

[50] Zhiheng Xu, Long Ru, Liang Xiang, and Qing Yang. Discovering user interest on twitter with a modified author-topic model. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '11, pages 422–429, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4513-4. doi: 10.1109/WI-IAT.2011.47. URL http://dx.doi.org/10.1109/WI-IAT.2011.47.

[51] Yu Wang, Eugene Agichtein, and Michele Benzi. Tm-lda: Efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 123–131, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339552. URL http://doi.acm.org/10.1145/2339530.2339552.

[52] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6. URL http://dl.acm.org/citation.cfm?id=1036843.1036902.

[53] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *In NIPS*, 2010.

[54] Saša Petrović, Miles Osborne, and Victor Lavrenko. Using paraphrases for improving first story detection in news and twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 338–346, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-20-6. URL http://dl.acm.org/citation.cfm?id=2382029.2382072.

[55] K. C. Wang. A Suggestion on the Detection of the Neutrino. *Physical Review*, 61: 97–97, January 1942. doi: 10.1103/PhysRev.61.97.

[56] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[57] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 245–250, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502546. URL http://doi.acm.org/10.1145/502512.502546.

[58] Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. Randomized algorithms and nlp: Using locality sensitive hash function for high speed noun clustering. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 622–629, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219917. URL http://dx.doi.org/10.3115/1219840.1219917.

[59] Ferhan Ture, Tamer Elsayed, and Jimmy Lin. No free lunch: Brute force vs. locality-sensitive hashing for cross-lingual pairwise similarity. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 943–952, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010042. URL http://doi.acm.org/10.1145/2009916.2010042.

[60] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[61] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng . P. Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.

[62] Abdelghani Bellaachia and Mohammed Al-Dhelaan. Ne-rank: A novel graph-based keyphrase extraction in twitter. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 372–379. IEEE Computer Society, 2012.

[63] Ou Jin, Nathan N. Liu, Kai Zhao, Yong Yu, and Qiang Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 775–784. ACM, 2011.

[64] Yuheng Hu, Ajita John, Dorée Duncan Seligmann, and Fei Wang. What were the tweets about? topical associations between public events and twitter feeds. In *ICWSM*, 2012.

[65] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng . P. Lim, and Xiaoming Li. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 379–388. Association for Computational Linguistics, 2011.

[66] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, volume 16, 2003.

[67] Ryan Prescott Adams, Zoubin Ghahramani, and Michael I. Jordan. Tree-structured stick breaking for hierarchical data. In *NIPS*, pages 19–27, 2010.

[68] Joon Hee Kim, Dongwoo Kim, Suin Kim, and Alice Oh. Modeling topic hierarchies with the recursive chinese restaurant process. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 783–792. ACM, 2012.

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

[70] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[71] David Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII1983*, pages 1–198, 1985.

[72] Zhiheng Xu, Rong Lu, Liang Xiang, and Qing Yang. Discovering user interest on twitter with a modified author-topic model. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 1, pages 422–429. IEEE, 2011.

[73] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[74] George Kingsley Zipf. *Selected studies of the principle of relative frequency in language*. Cambridge, Mass., Harvard university press, 1932.

[75] citeulike:9335608. Tweets2011 corpus. Online, 2011. URL https://sites.google.com/site/microblogtrack/2011-guidelines.

[76] Scott Deerwester. Improving information retrieval with latent semantic indexing. 1988.

[77] Inderjit S. Dhillon and Suvrit Sra. Generalized nonnegative matrix approximations with bregman divergences. In *NIPS*, volume 18, 2005.

[78] David Andrzejewski and Xiaojin Zhu. Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48. Association for Computational Linguistics, 2009.

[79] Svetlana Bodrunova, Sergei Koltsov, Olessia Koltsova, Sergey Nikolenko, and Anastasia Shimorina. Interval semi-supervised lda: Classifying needles in a haystack. In *Advances in Artificial Intelligence and Its Applications*, pages 265–274. Springer, 2013.

[80] Xian-Ling . L. Mao, Zhao-Yan . Y. Ming, Tat-Seng . S. Chua, Si Li, Hongfei Yan, and Xiaoming Li. Sshlda: a semi-supervised hierarchical topic model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 800–809. Association for Computational Linguistics, 2012.

[81] David Newman, Sarvnaz Karimi, Lawrence Cavedon, Judy Kay, Paul Thomas, and Andrew Trotman. External evaluation of topic models. In *Australasian Document Computing Symposium (ADCS)*, pages 1–8. School of Information Technologies, University of Sydney, 2009.

[82] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[83] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 215–224. ACM, 2010.

[84] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal, The*, 27(3):379–423, July 1948. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1948.tb01338.x.